

Intellio **C320** *Turbo*

**Multiport Controller for
Desktop and Rackmount**

An Advanced Professional
Intelligent Multiport System
for IBM PC/AT and
Compatible System

May 1998 (3rd Edition)

All Rights Reserved

Manual Organization

This manual is composed of eight chapters and three indices.

Chapter 1 Introduction

This chapter describes the features and specifications for *Intellio C320 Turbo*.

Chapter 2 Hardware Installation

This chapter describes the hardware installation for *Intellio C320 Turbo*.

Chapter 3 For Windows NT Users

This chapter describes the software installation, configuration, driver upgrade and removal for *Intellio C320 Turbo*. Also *Pcomm*, the serial comm tool and finally the troubleshooting specific for Windows NT.

Chapter 4 For Windows 95 Users

This chapter describes the software installation, configuration, driver upgrade and removal for *Intellio C320 Turbo*. Also *Pcomm*, the serial comm tool and finally the troubleshooting specific for Windows 95.

Chapter 5 PComm

This chapter roughly describes the installation for *Pcomm* under Windows NT/95, in addition to the library and utilities.

Chapter 6 For UNIX Users

This chapter describes the software installation, administrations including configuration, monitoring and removal for *Intellio C320 Turbo*. Also programming and baud rate setting issues and finally the troubleshooting specific for UNIX.

Chapter 7 For DOS Users

This chapter describes the software installation, setup and driver loading/unloading removal for *Intellio C320 Turbo*. Also API-232, the programming library and utilities such as Data Scope.

Chapter 8 For Windows 3.x Users

This chapter describes the software installation for *Intellio C320 Turbo*. And then details the Windows-compatible COMM driver and the Proprietary Windows DLL driver, respectively.

Appendix A Troubleshooting

This appendix describes the FAQs and possible answers for *Intellio C320 Turbo*.

Appendix B Technical Reference

This appendix describes the Dual-port memory, UART, pinouts and cable wiring for *Intellio C320 Turbo*.

Appendix C Extended UNIX ioctl() Commands

This chapter describes the MOXA-supported UNIX `ioctl()` commands for UNIX programming.

Copy

This documentation is copyrighted by Moxa Technologies Co., Ltd. All rights are reserved. Moxa Technologies reserves the right to make improvements to the products described in this manual at any time without notice.

Information provided in this manual is intended to be accurate and reliable. However, Moxa Technologies assumes no responsibility for its use, nor for any infringements of rights of the fourth parties which may result from its use.

MOXA is a registered trademark of Moxa Technologies Co. Ltd. The other trademarks in this manual belong to their manufacturers individually.

Turn PC into Online-Access-Server.

Moxa Headquarters

Tel: +886-2-8665-6373
Fax: +886-2-8665-6372

www.moxa.com

Moxa Tech USA (CA)

Tel: (408)734-2224
Fax: (408)734-4442

support@moxa.com.tw

Moxa Internet



Customer's satisfaction is always our number one concern. To ensure customers get the full benefit of our services, Moxa Internet Services (MIS) have been built for technical support, product inquiry, new driver upgrade, etc.

The followings are the services we provide.

E-mail for technical support

address: support@moxa.com.tw

Ftp site for free driver upgrade

address: [ftp.moxa.com](ftp://ftp.moxa.com) or [ftp.moxa.com.tw](ftp://ftp.moxa.com.tw)

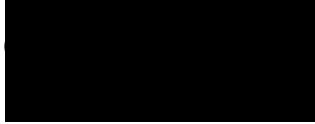
user ID: ftp

password: your_email_address

World Wide Web (WWW) for product info.

address: www.moxa.com or www.moxa.com.tw

Table



Chapter 1	Introduction	1
1.1	Overview	1
1.2	Features and Specifications	3
1.3	Check List.....	8
1.4	Installation Guide.....	8
Chapter 2	Hardware Installation	11
2.1	Installation of Control Board	11
2.2	Installation of External Modules.....	12
2.2.1	<i>Installation of CPU and UART Modules(Desktop Option).....</i>	<i>13</i>
2.2.2	<i>Installation of Basic and Extensive Modules(Rackmount Option).....</i>	<i>16</i>
2.3	LED Indicators and Operation.....	19
2.4	Mix of UART Modules	21
Chapter 3	For Windows NT Users.....	23
3.1	Driver Installation.....	23
3.2	Board and Port Configuration.....	24
3.3	Board Initialization Status Checking	25
3.4	Driver Upgrade/Removal.....	26
3.5	MOXA Serial Comm Tool: <i>PComm</i>	26
3.6	Troubleshooting by Error (Event) Messages	26
Chapter 4	For Windows 95 Users.....	29

4.1	Driver Installation.....	29
4.2	Board and Port Configuration.....	29
4.3	Board Initialization Status Checking	31
4.4	Driver Upgrade/Removal.....	31
4.5	MOXA Serial Comm Tool: <i>PComm</i>	31
4.6	Troubleshooting by Error Messages.....	31
Chapter 5	<i>PComm</i>.....	35
5.1	Installation	35
5.2	Programming Library	35
5.3	Utilities	35
Chapter 6	For UNIX Users	37
6.1	Driver Installation.....	37
6.2	MOXA TTY Device Names	39
6.3	Baud Rate Settings.....	42
6.4	Administrations.....	42
6.5	Setting MOXA Ports to Terminal	45
6.6	Programming the MOXA Ports	46
6.7	Troubleshooting.....	46
Chapter 7	For DOS Users	49
7.1	Driver Installation, Setup, Loading and Unloading.....	49
7.2	MOXA Serial Port Names	51
7.3	Programming Library	52
7.4	Data Scope	53

Chapter 8	For Windows 3.x Users	55
8.1	Driver Installation.....	55
8.2	MOXA Windows-compatible COMM Driver.....	56
8.3	MOXA Proprietary Windows DLL Driver.....	58
Appendix A	Troubleshooting	61
Appendix B	Technical Reference.....	67
B.1	Dual-Ported RAM.....	67
B.2	TI550C in UART/Basic/Extensive Modules	68
B.3	Cable Wiring.....	68
<i>B.3.1</i>	<i>RS-232/422 Cable Wiring for UART Modules (Desktop Option)</i>	<i>69</i>
<i>B.3.2</i>	<i>RS-232 Cable Wiring for Basic/Extensive Module (Rackmount Option).....</i>	<i>74</i>
<i>B.3.3</i>	<i>RS-232 Test Cable Wiring for Basic/Extensive Module (Rackmount Option)..</i>	<i>76</i>
B.4	Control Board to CPU/Basic Module Cable Pinouts	76
Appendix C	Extended UNIX ioctl() Commands	78

Chapter 1 Introduction

This manual covers both the hardware and software installation and configuration of **C320 Turbo**, which is the member of the *Intellio* Family. In addition, most powerful serial comm tool, *PComm* and **API-232**, are introduced. More over, troubleshooting and technical reference will help you solve your problem quickly. Finally, a summary installation guide is available in Section 1.4 to cut your installation time.

1.1 Overview

Excellent features of C320 *Turbo* are described as follows, which make C320 *Turbo* the best choice for medium to large-scale data communication systems.

Intellio - The Intelligent 8-32 Port Serial I/O Solution

The *Intellio* **C320 Turbo** is a dual-CPU high-speed expandable intelligent multi-serial communication controller which aims to gain higher speed, achieve better performance and lessen load of the host system in 8 to 32 serial port data communication solutions.

Port Expandable

The C320 *Turbo* is composed of a Control Board, a DB25 to DB25 cable, some external modules. The C320 *Turbo* supports expandability from 8 to 32 ports for a single PC slot. Maximum of **128** ports could be used in a ISA/EISA 386/486/Pentium based PC with four C320 *Turbo* installed.

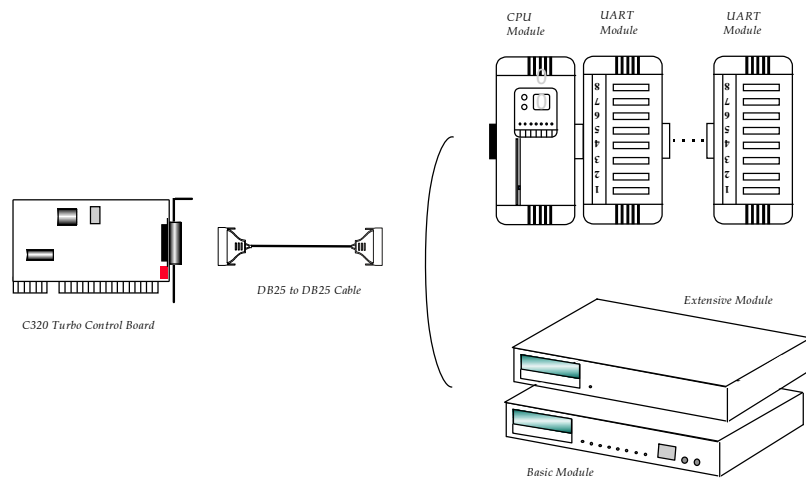
Distance Extendible

Normally, a standard 2-meter DB25 to DB25 cable with 25 signal pins links the Control Board to the external module. However, for the purpose of extending the distance between Control Board and external modules to above **2 m** and up to **100 m (328 ft)** or improving power insufficiency problem of the PC host, external power and the cable specially fabricated with only 10 signal pins should be used as illustrated in both

Chapter 2 and Appendix B.4.

Status Indicator

The status of the communication lines is displayed on a series of diagnostic LED indicators on the front panel of the external module, including TxD, RxD, DTR, DSR, RTS, CTS, DCD.



Versatile Connection Options

There are two options, **Desktop** and **Rackmount**, for the external modules to offer various connection solutions. For Desktop option, CPU Module and UART Modules are provided. For Rackmount option, Basic Module and Extensive Modules are provided with rack mount capability. Please select one of the options that meets the application needs.

Operating System Support

It supports most popular O.S. platforms like Windows NT, Windows 95, Linux, SCO UNIX/OpenServer, UNIX SVR4.2, DOS and Windows 3.x. **MOXA device drivers** feature easy installation, configuration and better performance. In this manual, chapters for MOXA **Windows NT**, **Windows 95**, **UNIX**, **DOS** and **Windows 3.1** device drivers

are included. For other systems not mentioned, please contact Moxa dealer/distributor or Moxa or visit the MOXA **Web** site for more information about newly available device drivers.

MOXA Serial Comm Tool

For application development, MOXA provides an easy-to-use while powerful serial communication library under Windows NT/95 (*PComm*) and Windows 3.x/DOS (**API-232**). Users can use this library to develop your own applications using Visual C++, Visual Basic, Borland C++, Borland Delphi, Microsoft C, Turbo C, Assembly, QuickBASIC, Turbo Pascal, Clipper, etc. Utilities, such as Data Scope, monitor, terminal emulation, diagnostics, etc., are included for debugging or monitoring the communication status or terminal emulation or even file transferring.

Easy Installation

No matter hardware or software, installation are made as easy as possible. Please follow the installation guide to install and configure the hardware and the driver. Then, you can start to use C320 *Turbo* to transmit/receive data to/from the connected devices, such as terminals, modems and printers, with a ready-made or self-written application program.

Detailed Information

At the end of this manual, common troubleshooting and technical reference are appended for problem solving and cable wiring.

1.2 Features and Specifications

Features

- ✓ High speed serial communication»Up to 460.8 Kbps
- ✓ Low host CPU's overhead»Dual RISC processor architecture
- ✓ Reliability»On-chip hardware flow control guarantees no data loss
- ✓ Modular expandability»Easy to add ports for a single PC slot
- ✓ Supports full communication status display for each port
- ✓ Long range extensibility»Easy for long distance cable layout

- ✓ Rack mountable»Industrial standard
- ✓ Supports most popular OSs
- ✓ Easy software upgrading»Firmware downloading
- ✓ Friendly user interface for configuration and utilities
- ✓ Powerful programming library and illustrative examples

Specifications

System

- ✓ Up to 32 serial ports per Control Board
- ✓ Up to 4 Control Boards per PC system for a total of 128 ports
- ✓ Supports device drivers for most popular operating systems. See the driver support list below:

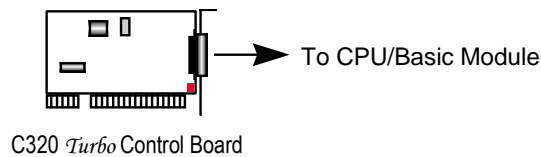
	<i>C320 Turbo</i>
Windows NT	✓
Windows 95	✓
Windows 3.1	✓
DOS	✓
SCO UNIX/OpenServer	✓
UNIX SVR4.2	✓
SCO XENIX	✓
OS/2	R
Linux	R
QNX	R

✓ : Driver supported by Moxa and shipped with product

R : Driver supported by Moxa but sent by request or download from MOXA FTP

Control Board

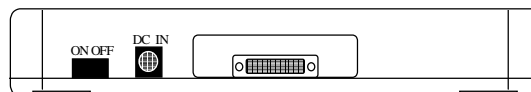
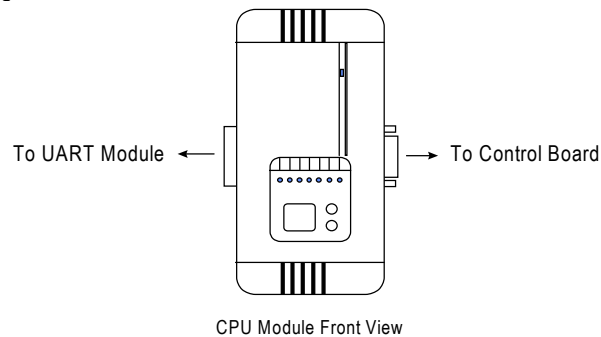
- ✓ 40 MHz TMS320C52 processor
- ✓ 512 KB dual-ported RAM, maps to just 16K Bytes system RAM
- ✓ CPU/Basic Module can be located up to 100 m (328 ft) away from the host computer
- ✓ Six possible base addresses from C8000 to DC000 (hex), addresses can be above 1 MB for UNIX systems
- ✓ Selectable IRQ: 2, 3, 4, 5, 7, 10, 11, 12, or 15
- ✓ Half-size board
- ✓ FCC Class A



External Modules for Desktop Option: CPU Module and UART Module

CPU Module (for Desktop option)

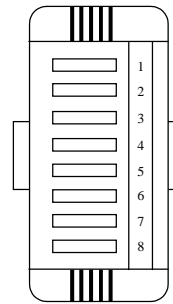
- ✓ 40 MHz TMS320C52 processor
- ✓ 16 KB ROM and 32 K Words RAM
- ✓ Diagnostic LEDs show full status of each serial port, including TxD, RxD, DTR, DSR, RTS, CTS, DCD
- ✓ Optional power adapter for use when the distance between Control Board and CPU Module is longer than 2 meters. In this case, a 10-signal-pin DB25 to DB25 cable must be used between Control Board and CPU Module.
- ✓ Power switch
- ✓ FCC Class A



CPU Module Side View

UART Module (for Desktop option)

- ✓ 8 RS-232/RS-422 serial ports
- ✓ TI550C UART chip with 32 bytes FIFO (Tx+Rx) and on-chip hardware flow control
- ✓ Each UART chip drives one serial port (Max. 460.8 Kbps)
- ✓ Up to 4 (at least one) modules can be connected to one Control Board
- ✓ RS-232 (DB25 female/male connectors) : TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND (Full modem control signals)
- ✓ RS-422 (DB25 female connectors) : TxD+/-, RxD+/-, CTS+/-, RTS+/-, GND
- ✓ Fixing Kit with 2 metal plates and 4 screws
- ✓ FCC Class A

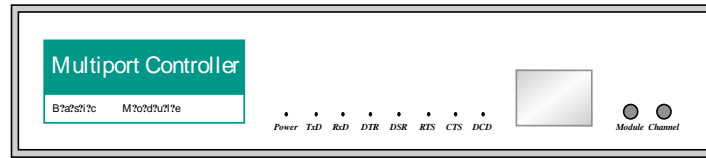


UART Module

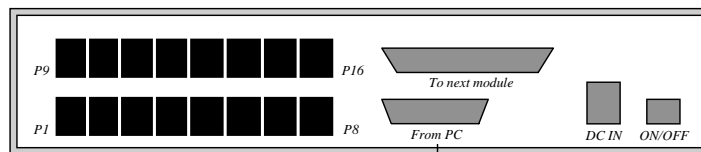
External Modules for Rackmount Option: Basic Module and Extensive Module:**Basic Module** (for Rackmount option)

- ✓ 8 or 16 serial ports
- ✓ 40MHz TMS320C52 processor
- ✓ 16KB ROM and 32K Words RAM
- ✓ Diagnostics LEDs show full status of each serial port, including TxD, RxD, DTR, DSR, RTS, CTS, DCD
- ✓ TI550C UART chip with 32 bytes FIFO (Tx+Rx) and on-chip hardware flow control
- ✓ Each UART chip drives one serial port (Max. 460.8K bps)
- ✓ Full modem control (RS-232 interface)
- ✓ RJ-45 connector for each serial port
- ✓ Optional power adapter for use when the distance between Control Board and Basic Module is longer than 2 meters. In this case, a 10-signal-pin DB25 to DB25 cable must be used between Control Board and Basic Module.

✓ Power switch



Basic Module (Front View)

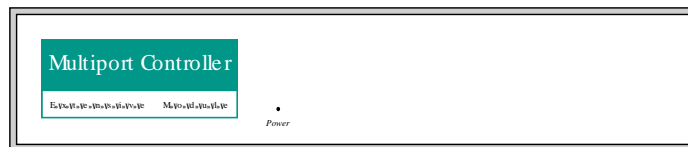


To Control Board

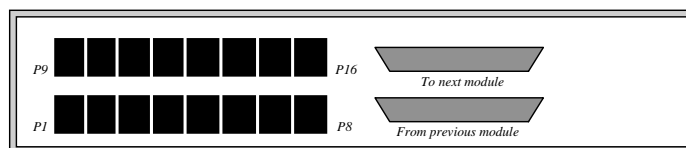
Basic Module (Rear View)

Extensive Module (for Rackmount option)

- ✓ 8 or 16 RS-232 serial ports
- ✓ TI550C UART chip with 32 bytes FIFO (Tx+Rx) and on-chip hardware flow control
- ✓ Each UART chip drives one serial port (Max. 460.8 Kbps)
- ✓ Full modem control (RS-232 interface)
- ✓ RJ-45 connector for each serial port



Extensive Module (Front View)



Extensive Module (Rear View)

1.3 Check List

Upon unpacking C320 *Turbo* package, the following items are included:

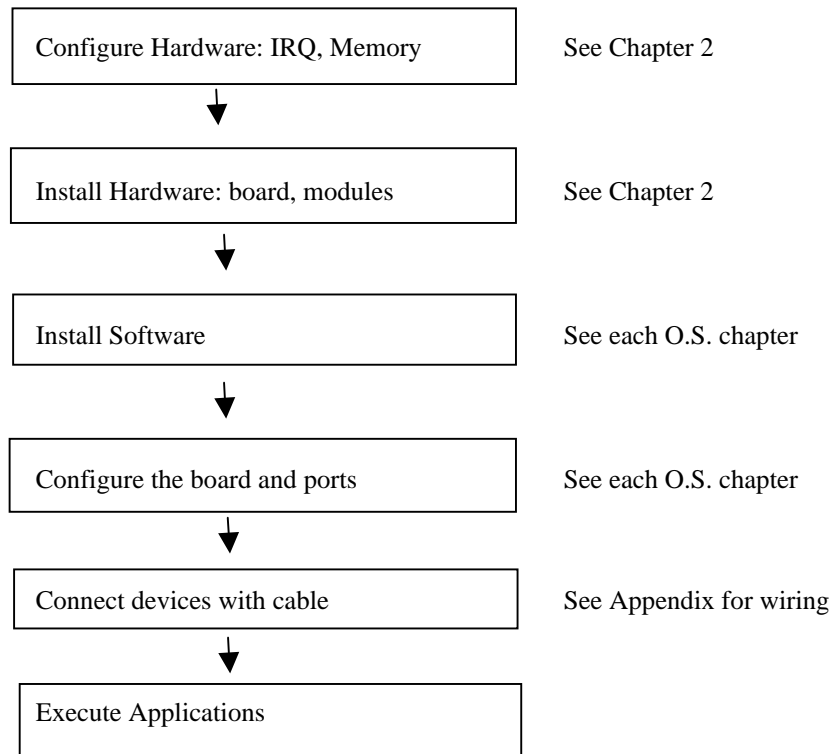
- ✓ One C320 *Turbo* Control Board
- ✓ One CPU Module or Basic Module
- ✓ One 2-meter DB25 to DB25 cable for connecting Control Board and CPU/Basic Module. This item may not be needed if long range extension kit is purchased
- ✓ For Desktop option, at least one and up to four 8-port RS-232/RS-422 female/male UART Module(s). For Rackmount option, up to 3 Extensive Module(s)
- ✓ For Rackmount option, DB-37 to DB-37 cable(s) for connecting Basic/Extensive Modules, if Extensive Module is used
- ✓ For Rackmount option, one 1.5-meter RJ-45 to male DB25 RS-232 cable for testing
- ✓ Device driver diskettes: Windows NT/95 »ë 1, DOS/Windows 3.x »ë 1, UNIX/XENIX»ë1
- ✓ MOXA Serial Comm Tool diskette *PCComm»ë1*
- ✓ This C320 *Turbo* User's Manual
- ✓ API-232 Manual
- ✓ Fixing Kit for Desktop option. Or Rack Mount Kit for Rackmount option.

The following item may be included if long range extension is needed:

- ✓ Long range extension kit
 - »öit A power adapter for CPU/Basic Module, 90-240V AC auto-select
 - »öit A DB25 to DB25 cable which contains only 10 signal pins for connecting Control Board and CPU/Basic Module

1.4 Installation Guide

This section gives a brief summary of how to install the C320 *Turbo* under each supported operating system. Installation is simple and involves the following stages:



Chapter 2 Hardware Installation

The C320 *Turbo* hardware installation consists of Control Board installation (Section 2.1) and external module installation (Section 2.2).

2.1 Installation of Control Board

Step 1: Turn off the PC's power.

Warning!	<i>Make sure your system is switched off before you start installing any board. If you don't, you may risk damaging your system and the board.</i>
-----------------	--

Step 2: Remove the PC's cover.

Step 3: Choose an IRQ number which is not used by other interface boards. There are 9 IRQ numbers you can choose from at jumper JP1. If you are installing more than one C320 *Turbo* or C218 Series board, their IRQ number must be set to the same number, i.e., **multiple boards share only one IRQ**.

JP1 (ON: install jumper) (*: default)									IRQ
1	2	3	4	5	6	7	8	9	
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	4
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	5
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	7
OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	10*
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	11
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	12
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	15

Step 4: Choose a base address (occupying 16KB) which is not used by expansion memory or other add-on boards, such as SCSI adapter.

If you are installing more than one C320 *Turbo* (or C218 Series) board, **each**

board must have a unique memory address.

Set the switch SW1 to the desired address as shown in the table below. SW1 can be accessed without removing your PC's cover after the board has been installed. Note that all MOXA drivers support the 6 base addresses below 1 MB, which are listed below. For UNIX users, base addresses above 1 MB and below 16 MB are available, please refer to Appendix B.1.

SW1 Dip switch settings (*: default)

1	2	3	4	5	6
ON	OFF	ON	ON	ON	ON
OFF	OFF	ON	ON	ON	ON
ON	ON	OFF	ON	ON	ON
OFF	ON	OFF	ON	ON	ON
ON	OFF	OFF	ON	ON	ON
OFF	OFF	OFF	ON	ON	ON

Base Address
C8000
CC000
D0000
D4000
D8000
DC000*

Note!

Make a note of the selected IRQ number and base memory address for software driver configuration use later.

Step 5: Remove the slot cover bracket if present.

Step 6: Plug the C320 *Turbo* Control Board firmly into any unused 16-bit slot.

Step 7: Fasten the holding screw to make the Control Board fixed.

Step 8: Replace the system cover.

Now the installation of the Control Board is complete. Continue to install the external modules.

2.2 Installation of External Modules

There are two options for the installation of external modules: CPU and UART Modules for Desktop option (Section 2.2.1) or Basic and Extensive Modules for Rackmount option (Section 2.2.2).

Normally, a standard 2-meter DB25 to DB25 cable with 25 signal pins links the Control Board to the external module. However, to extend the distance between Control Board and external module to above **2 m** and up to **100 m (328 ft)** or to improve power insufficiency problem of the PC host, **external power plugged in the CPU/Basic Modules along with a cable specially fabricated with only 10 signal pins** should be used as illustrated in the next two sub-sections and in Appendix B.4 as well. Otherwise, power degradation which comes from longer cable or power insufficiency of PC host will cause system failure.

2.2.1 Installation of CPU and UART Modules(Desktop Option)

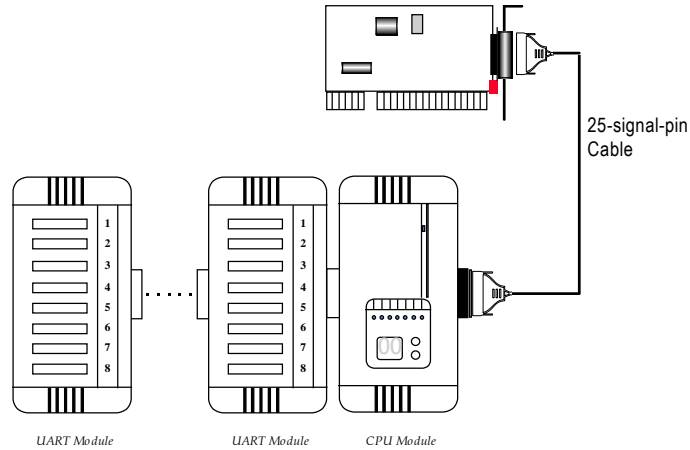
Step 9: Connect the C320 *Turbo* Control Board to the CPU Module with the shipped DB25 to DB25 cable as shown in the following picture.

If range extension or external power is required, please use 10-signal-pin cable coming with long range extension kit, instead. Thus, there are two types installation: **without power adapter** and **with power adapter**, which are described as follows:

Without power adapter (for normal condition)

In most cases, users need not a power adapter.

- A: Set the CPU Module power switch to the **OFF** position. This is absolutely necessary when installing or removing the cable, the CPU Module or the UART Module(s). Power should not be switched on until you installed all components.
- B: Plug the DB25 male end of the shipped 2-meter 25-signal-pin cable into the connector on the rear panel of the C320 *Turbo* Control Board. The pinouts of the cable are shown in Appendix B.4 Type 1.

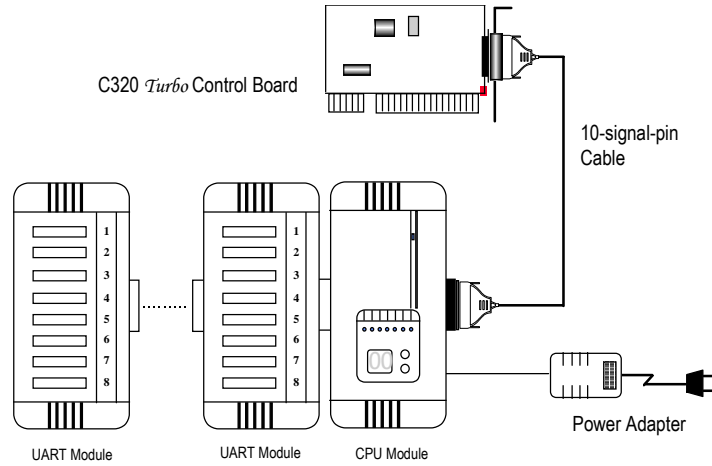


- C: Plug the other DB25 female end into the CPU Module's DB25 connector.

With power adapter (for range extension or external power condition)

In case of range extension or external power requirement, users need a power adapter.

- A: Set the CPU Module power switch to the **OFF** position. This is absolutely necessary when installing or removing the cable, the CPU Module or the UART Module(s). Power should not be switched on until you installed all components.
- B: Plug the DB25 male end of the shipped 2-meter 10-signal-pin cable (the one comes with long range extension kit or the one fabricated according to the pinouts in Appendix B.4 Type 2) into the connector on the rear panel of the C320 *Turbo* Control Board.



In fact, users can hook up a 25-signal-pin cable to the existing 10-signal-pin cable, forming a 2-segment cable, to extend the connection distance up to 100 meters (328 ft). Alternatively, make your 100 meter 10-signal-pin cable according to the precise pinouts described in Appendix B.4 Type 2.

- C: Plug the other DB25 female end into the CPU Module's DB25 connector.

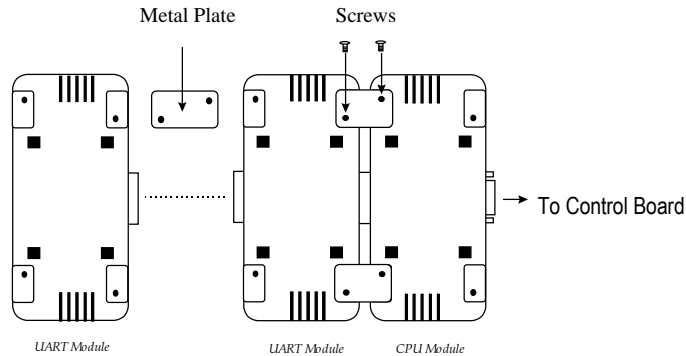
Warning!

Do not use a 25-signal-pin cable to connect the C320 Turbo Control Board to the CPU Module when using the power adapter as this will cause power crash. (One power comes from the power adapter while the other power comes from the C320 Turbo Control Board.)

- D: Connect the power adapter to the CPU Module. Keep the CPU Module's power switch in the OFF position. If UART Module(s) is (are) also required, keep the CPU Module's power switch in the OFF position until all necessary UART Module(s) is (are) installed.
- E: Install the power adapter to a power source, either 110V or 220V AC.

Adjustment to the AC power specs is done automatically.

Step 10: Connect the first UART Module to the CPU Module. Connect the second UART Module to the first one if necessary and so on.



For better fixation of modules, **Fixing Kit** is available and see the bottom view of modules below to install.

Step 11: After making sure that each component has been correctly installed, you are recommended to power on the CPU Module first and then power on the PC system secondly.

Now the installation of the external CPU/UART modules is complete. Continue to install the software driver explained in each O.S. chapter.

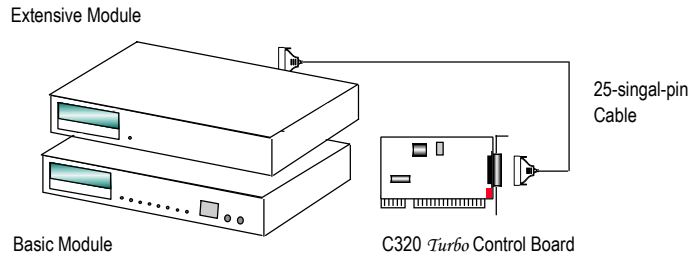
2.2.2 Installation of Basic and Extensive Modules(Rackmount Option)

Step 9: Connect the C320 *Turbo* Control Board to the Basic Module with the shipped DB25 to DB25 cable as shown in the following picture.

If range extension or external power is required, please use 10-signal-pin cable coming with long range extension kit, instead. Thus, there are two types installation: **without power adapter** and **with power adapter**, which are described as follows:

Without power adapter (for normal condition)

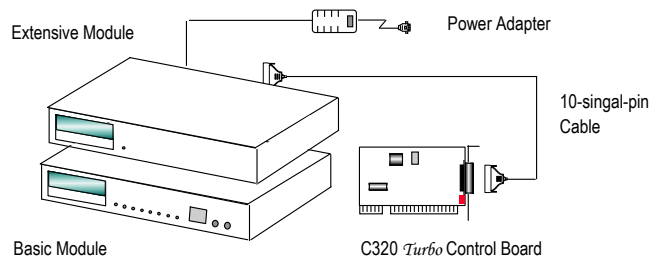
In most cases, users need not a power adapter.



- A: Set the Basic Module power switch to the **OFF** position. This is absolutely necessary when installing or removing the cable, the Basic Module or the Extensive Module(s). Power should not be switched on until you installed all components.
- B: Plug the DB25 male end of the shipped 2-meter 25-signal-pin cable into the connector on the rear panel of the C320 *Turbo* Control Board. The pinouts of the cable are shown in Appendix B.4 Type 1.
- C: Plug the other DB25 female end into the Basic Module's DB25 connector.

With power adapter (for range extension or external power condition)

- A: Set the Basic Module power switch to the **OFF** position. This is absolutely necessary when installing or removing the cable, the Basic Module or the Extensive Module(s). Power should not be switched on until you installed all components.



- B: Plug the DB25 male end of the shipped 2-meter 10-signal-pin cable (the one comes with long range extension kit or the one fabricated according to the pinouts in Appendix B.4 Type 2) into the connector on the rear panel of the C320 *Turbo* Control Board.

In fact, users can hook up a 25-signal-pin cable to the existing 10-signal-pin cable, forming a 2-segment cable, to extend the connection distance up to 100 meters (328ft). Alternatively, make your 100 meter 10-signal-pin cable according to the precise pinouts described in Appendix B.4 Type2.

- C: Plug the other DB25 female end into the Basic Module's DB25 connector.

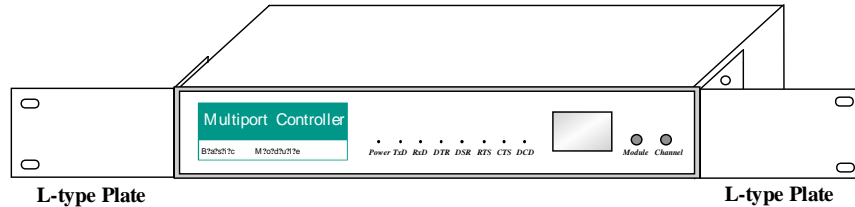
Warning!	<i>Do not use a 25-signal-pin cable to connect the C320 Turbo Control Board to the Basic Module when using the power adapter as this will cause power crash. (One power comes from the power adapter while the other power comes from the C320 Turbo Control Board)</i>
-----------------	---

- D: Connect the power adapter to the Basic Module. Keep the Basic Module's power switch in the OFF position. If Extensive Module(s) is(are) also required, keep the Basic Module's power switch in the OFF position until all necessary Extensive Module(s) is (are) installed.

- E: Install the power adapter to a power source, either 110V or 220V AC. Adjustment to the AC power specs is done automatically.

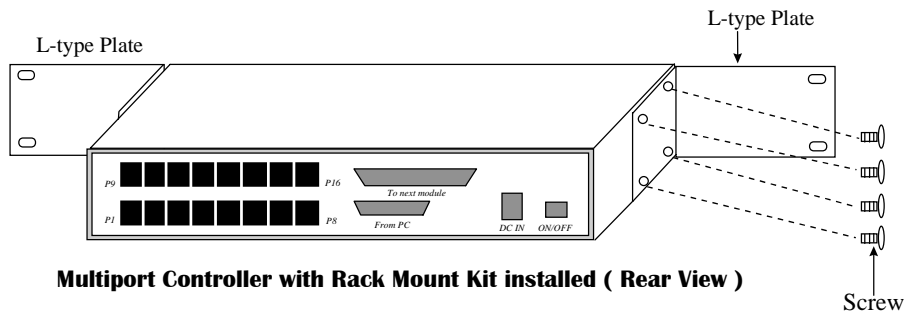
Step 10: If one more Extensive Module is required, plug the DB37 to DB37 male end of the shipped 1-meter cable into the DB37 female connector on the rear panel of the Basic Module and the other end of the cable into the DB37 male connector on the rear panel of the Extensive Module. If more Extensive Module is needed, connect the next Extensive Module to the previous one as described above.

To mount the module(s) on the industrial standard 19" rack, **Rack Mount Kit**, including two L-type plates and eight screws, should be applied.



Multiport Controller with Rack Mount Kit installed (Front View)

Step 11: After making sure that each component has been correctly installed, you are



Multiport Controller with Rack Mount Kit installed (Rear View)

recommended to power on the Basic Module first and then power on the PC system secondly.

Now the installation of the external Basic/Extensive module is complete. Continue to install the software driver explained in each O.S. chapter.

2.3 LED Indicators and Operation

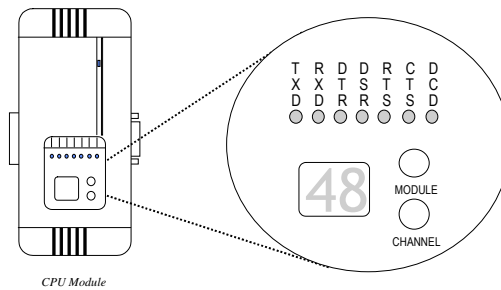
After completing the installation and powering on the CPU/Basic Module and the PC system, check the two-digit LED display on the CPU/Basic Module. These LEDs show the results of the system self-diagnostic tests which are run by the CPU/Basic Module after startup.

The CPU/Basic Module will first test the ROM and RAM of itself, and then UART/Extensive Module(s) if present. If any error is found, the LED display will show

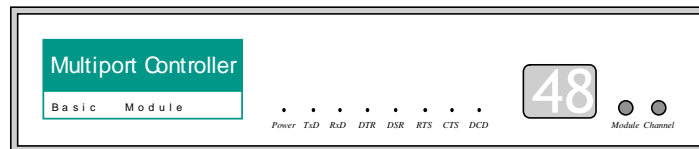
one of the messages described in Appendix A Troubleshooting.

If the first test passed, the CPU/Basic Module will then display "Ld" waiting for loading firmware from the C320 *Turbo* Control Board. After loading the firmware, the CPU Module will scan for the number of UART Modules or the number of ports available. The LED will show the last accessible port.

For CPU Module, the left digit shows the UART Module number and the right digit shows the last port number within a UART Module. The UART Module closest to the CPU Module is of number 1. The next module is of number 2, and so on. For example, if "48" is displayed, it means that the last accessible port is the eighth port of the fourth UART Module.



For Basic Module, the left digit shows the number of 8-port unit that configured (if continuous 8 ports are considered as an 8-port unit) and the right digit shows the last port number within an 8-port unit. For example, if "48" is displayed, it means that the last accessible port is the eighth port of the fourth 8-port unit.



Basic Module

To see a particular port's line status, users can keep pressing Module Button and Channel (Port) Button till the desired port is shown on LED display, then look at the seven indicators TxD, RxD, DTR, DSR, RTS, CTS, and DCD. This provides a convenient diagnostic ways for C320 *Turbo*. Normally, DTR and RTS indicators will light on when local port is opened while DSR and CTS indicators will light on when remote port is opened. TxD indicator will light on if local port is transmitting data and

RxD indicator will light on if local port is receiving data (or remote port is transmitting data). DCD indicator is mostly useful to detect phone line carrier when a modem is connected to the port.

2.4 Mix of UART Modules

UART Modules with various functions are produced for different requirements, such as RS-232/RS-422, female/male DB25 connector, isolation protection or surge protection.

No matter what interfaces they use, any two UART Modules could be mixed (connected) together freely depending on needs. For example, users may put 2 male RS-232 UART Modules and 2 female RS-422 UART Modules together for application consideration.

Chapter 3 For Windows NT Users

Windows NT supports up to **256** serial ports, from **COM1** to **COM256**. To fully integrate the advanced features of Windows NT, multi-process and multi-thread, pure 32-bit Windows NT device drivers are developed for C218 *Plus* and other MOXA multiport boards. The drivers conform to Win32 COMM API standard.

In this chapter, software driver installation, configuration and driver upgrade/removal procedure is described.

Once the board and driver are installed and the system restart successfully, users can execute ready-made applications such as HyperTerminal to transmit/receive data or Remote Access Service to provide dial-up networking capability.

If it is necessary for you to develop your own applications, we strongly recommend the easy but powerful *PComm*, serial comm tool, which includes library for easy programming and useful utilities for diagnostics and monitor.

Finally, Windows NT-specific troubleshooting is included for quick problem solving.

3.1 Driver Installation

The following is the procedure for installing the C320 *Turbo* device driver under Windows NT 4.0 (or NT 3.51):

1. Please Login NT as **Administrator**.
2. From [**Control Panel**], select [**Network**] icon and then [**Adapters**] tab.
(From [**Control Panel**], select [**Network**] icon for NT 3.51.)
3. Click [**Add**] button, then [**Have Disk...**] button for "Select Network Adapter".
(Click [**Add Adapter...**] button and then "<Other> **Requires disk from manufacturer**" item for "Network Adapter Card:" for NT 3.51.)
4. Specify the exact path for the driver diskette, e.g. **A:\WINDOWS.NT**. Then

click [**Continue/OK**].

5. Select "**MOXA C218/C320 multiport board series**" in "Select OEM Option" dialog, then click [**OK**] to start installation.
6. The "**MOXA C218/C320 Series Configuration Panel**" dialog will pop up. Please go directly to step 3 in "Board and Port Configuration" Section for more configuration details.

When configuration is done, select [**OK**] for "Network Settings" dialog.

Up to now, users have already completed the whole installation, please restart system as prompted.

Note!

Please double check if all the C320 Turbo components: Control Board, link cable and external modules, are connected and fastened tightly in order to ensure the system and driver will start successfully later.

Once the board and driver are installed and the system restart successfully, users can execute any ready-made application such as HyperTerminal to transmit/receive data or Remote Access Service to provide dial-up networking capability.

Optionally, users may check the board initialization status after logging in the system. Refer to Section "Board Initialization Status Check" for details.

3.2 Board and Port Configuration

The following is the procedure for configuring the C320 *Turbo* driver under Windows NT 4.0 (or NT 3.51):

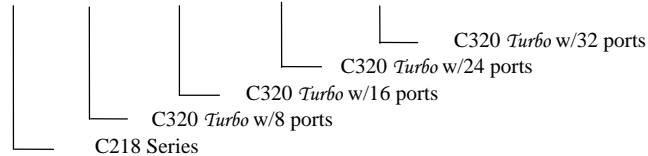
1. From [**Control Panel**], select [**Network**] icon and then [**Adapters**] tab.
(From [**Control Panel**], select [**Network**] icon for NT 3.51.)
2. Select "**MOXA C218/C320 Series Adapter**" item for "Installed Adapter

Cards:", then [**Configure/Properties**] button to start the "MOXA C218/C320 Series Configuration Panel" dialog.

3. When the configuration panel is popped up:

Click [**Add**] button to add a board. Click [**Remove**] button to remove a board. Click [**Properties**] button to set up a board with correct "Board Type", "Memory Bank", "IRQ" and "COM Number", where the possible parameters are as follows:

Board Type: C218, C3208, C32016, C32024, C32032



Memory Bank: C8000, CC000, D0000, D4000, D8000, DC000

IRQ: 2, 3, 4, 5, 7, 10, 11, 12, 15

COM Number: from COM1 to COM256. It is a way to map the physical ports to COM numbers. Specify the COM number of the first port and subsequent ports are mapped to continuous COM numbers. For instance, if first port is mapped to COM10, then second port is mapped to COM11 sequentially.

At most 4 boards of combination of C320 *Turbo* and C218 Series can be installed together as long as the memory resources are sufficient and available in a system. These boards share only one same IRQ.

4. Unless the system restarted, the latest board configuration will not take effect.

3.3 Board Initialization Status Checking

There are two alternative ways to find out if ports of the board are initialized successfully:

1. From [**Control Panel**], select [**Ports**] icon to check the port list and see if all configured ports are already added in the system correctly.

If no configured port shown in the port list, refer to Section "Troubleshooting by Error (Event) Messages" for solutions.

2. Enter [**Administrative**] group, then enter [**Event Viewer**] icon and then enter "**Application**" item to check for message like "MOXA C320 *Turbo* Serial I/O Board 1 : Addr(C8000) O.K." for each configured board.

If message is "MOXA C320 *Turbo* (C8000) not found!" or "MOXA C320 *Turbo* Serial I/O Board #1 Addr(C8000) : IRQ(10) test failure!", refer to Section "Troubleshooting by Error (Event) Messages" for solutions.

3.4 Driver Upgrade/Removal

To upgrade driver, remove the installed driver first and install the new one.

To remove driver,

1. From [**Control Panel**], select [**Network**] icon and then [**Adapters**] tab.
(From [**Control Panel**], select [**Network**] icon for NT 3.51.)
2. Select "**MOXA C218/C320 Series Adapter**" in adapter list, then [**Remove**] button to start removing the MOXA board.

3.5 MOXA Serial Comm Tool: *PComm*

PComm is a software package consisting of MOXA serial communication library and utilities for diagnostic and monitor as well as example programs under Windows NT/95. Please see Chapter "*PComm*" for details.

3.6 Troubleshooting by Error (Event) Messages

1. **MOXA C320 *Turbo* (XX000) not found !**

Please shutdown the system first, then refer to the corresponding problem and

solutions in Appendix "Troubleshooting".

2. MOXA C320 *Turbo* (XX000) CPU/Basic Module not found !

Please shutdown the system first, then refer to the corresponding problem and solutions in Appendix "Troubleshooting".

3. MOXA C320 *Turbo* (XX000) number of UART Modules mismatch !

Make sure that all the modules are fastened properly.

From [Control Panel], select [Network] icon, then [Adapters] tab, then select "MOXA C218/C320 Series Adapter" item for "Installed Adapter Cards:" and then [Configure/Properties] button to start the "MOXA C218/C320 Series Configuration Panel" dialog, make sure to connect the CPU Module with proper number of UART Module(s) as configured.

If the problem persists, please shutdown the system first, then refer to the corresponding problem in Appendix "Troubleshooting" for solutions.

4. MOXA C320 *Turbo* (XX000) number of ports mismatch !

Make sure that all the module(s) is(are) fastened properly and the ports installed are the same as configured.

From [Control Panel], select [Network] icon, then [Adapters] tab, then select "MOXA C218/C320 Adapter" item for "Installed Adapter Cards:" and then [Configure/Properties] button to start the "MOXA C218/C320 Series Configuration Panel" dialog, make sure to connect the CPU Module with proper number of UART Module(s) as configured.

If the problem persists, please shutdown the system first, then refer to the corresponding problem in Appendix "Troubleshooting" for solutions.

5. MOXA C320 *Turbo* Serial I/O Board Addr(C8000) : IRQ test failure!

This indicates MOXA board is found but the IRQ is not properly configured or conflicting with other adapter. Please:

- Make sure that both software and hardware IRQ setting do match.
- Make sure there is no conflict with other adapter's IRQ.
- Make sure IRQ is not used by PCI bus.

6. Windows NT system panic (blue screen).

The possible reason is IRQ or memory conflict with other adapters like LAN or SCSI or system BIOS. Please refer to the corresponding problem in Appendix A "Troubleshooting" for solutions.

Chapter 4 For Windows 95 Users

Windows 95 supports up to **128** serial ports, from **COM1** to **COM128**. To fully integrate the advanced features of Windows 95, multi-process and multi-thread, pure 32-bit Windows 95 virtual device port drivers (VxD) compliant with communication driver (VCOMM) are developed for C320 *Turbo* and other MOXA multiport boards. The drivers conform to Win32 COMM API standard.

In this chapter, software driver installation, configuration and upgrade/removal procedure are described. If it is necessary for you to develop your own applications, we strongly recommend the easy but powerful *PComm* Serial Comm Tool, which includes library for easy programming and useful utilities for diagnostics and monitor. Finally, Windows 95-specific troubleshooting is included.

4.1 Driver Installation

The following is the procedure for installing C320 *Turbo* under Windows 95:

1. Run **Setup95.exe** in the driver diskette.
2. Select "**MOXA C218/C320 Series Driver**" to install and configure the boards following the on-line instructions.

Please refer to the next Section "Board and Port Configuration" for more configuration details.

4.2 Board and Port Configuration

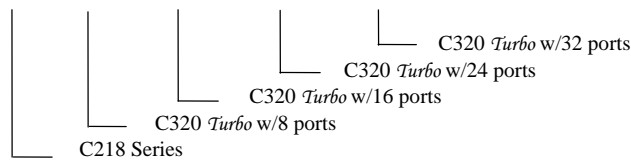
You will enter the configuration program when installing the device driver. Or you can click on the Taskbar **[Start]** button, then select **[Programs]** menu, then **[MOXA Utilities]** menu and then **[MOXA C218_C320 Configuration Panel]** icon.

When the configuration panel is popped up:
Click [**Add**] button to add a board.

Click [**Remove**] button to remove a board.

Click [**Properties**] button to set a board with correct "Board Type", "Memory Bank", "IRQ" and "COM Number", where the possible parameters are as follows:

Board Type: C218, C3208, C32016, C32024, C32032



Memory Bank: C8000, CC000, D0000, D4000, D8000, DC000.

IRQ: 2, 3, 4, 5, 7, 10, 11, 12, 15.

There are two ways to map the physical ports to COM numbers (COM1-128).

Sequential Port Mapping: Specify the COM number of the first port and subsequent ports are mapped to continuous COM numbers. For instance, if the first port is mapped to COM10, then the second port is mapped to COM11 sequentially.

Non-sequential Port Mapping: Specify the COM number for individual port. For instance, the second port can be out of sequence, say COM18, while the first port is mapped to COM10.

At most 4 boards of combination of C320 *Turbo* and C218 Series can be installed together as long as the memory resources are sufficient and available in a system. These boards share only one same IRQ.

Unless the system restarted, the latest configuration will not take effect.

Note!

Please double check if all the C320 Turbo components: Control Board, link cable and external modules, are connected and fastened tightly in order to make sure the driver will start normally later.

4.3 Board Initialization Status Checking

All the error conditions of the board, **during the initialization of the Windows 95**, will be popped up onto the screen. Otherwise, everything should be fine.

If message like "Base memory [C8000] is absent or occupied by others !" or "MOXA C320 *Turbo* [C8000] interrupt failure !" shows, refer to Section "Troubleshooting by Error Messages" for solutions.

Additionally, to make sure the board installation is successful, please run **Diagnostic** utility in *PComm*.

4.4 Driver Upgrade/Removal

To upgrade the driver, simply install the driver again.

To remove driver, just remove all the configured boards. Once the MOXA driver is installed, it exists permanently in Windows 95 like other adapters do. Removing changes nothing but that particular board's existence in Windows 95 Registry.

4.5 MOXA Serial Comm Tool: *PComm*

PComm is a software package consisting of MOXA serial communication library and utility for diagnostic as well as example programs under Windows NT/95. Please see Chapter "*PComm*" for details.

4.6 Troubleshooting by Error Messages

If problem still persists after this troubleshooting, please refer to the corresponding problem and solutions in Appendix "Troubleshooting".

1. Base memory [XX000] is absent or occupied by others!

To avoid this, please double check the board settings via clicking on the Taskbar [Start] button, then [Programs] menu, then [MOXA Utilities] menu and then

[MOXA C218_C320 Configuration Panel] icon for each board installed.

For more information, please refer to Appendix "Troubleshooting" - "MOXA board not found."

2. MOXA C320 *Turbo* download failure!

Please shutdown the system first, then refer to Appendix "Troubleshooting" - "CPU/Basic Module can not download firmware from Control Board."

3. MOXA C320 *Turbo* [XX000] number of UART Modules mismatch!

Make sure that all the module(s) is(are) fastened properly.

Click on the Taskbar [Start] button, then [Programs] menu, then [MOXA Utilities] menu and then [MOXA C218_C320 Configuration Panel] icon, and connect the CPU Module with proper number of UART Module(s) as configured.

4. MOXA C320 *Turbo* number of ports mismatch!

Make sure that all the Module(s) is(are) fastened properly and the ports installed are the same as configured.

Click on the Taskbar [Start] button, then [Programs] menu, then [MOXA Utilities] menu and then [MOXA C218_C320 Configuration Panel] icon, and connect the Basic Module with proper number of Extensive Module(s) as configured.

5. MOXA C320 *Turbo* CPU/Basic Module not found!

Please shutdown the system first, then refer to Appendix "Troubleshooting" - "CPU/Basic Module not found."

6. MOXA C320 *Turbo* [XX000] interrupt failure!

This indicates MOXA board is found but the IRQ is not properly configured or conflicting with other adapter. Please:

- Make sure that both software and hardware IRQ setting do match.
- Make sure there is no conflict with other adapter's IRQ.

- Make sure IRQ is free and not used by PCI bus.

7. Mx2win95.cod file not found!**Mx2win95.cod file length is wrong!****Mx2win95.cod file data content error!**

It means something is wrong with driver installation or the file has been deleted accidentally or corrupted somehow. To recover, simply reinstall the driver.

Chapter 5

PComm

PComm, the professional serial comm tool for PC, is a software package under Windows NT/95, which consists of powerful serial communication library for easy programming in most popular languages, useful utilities such as diagnostic and monitor, illustrative example programs and comprehensive on-line documents.

The serial communication library is especially for users who develop a system for data communication, remote access, data acquisition or industrial control in the Windows NT/95 environment, which offers an easier solution compared with the more complex Windows Win32 COMM API.

5.1 Installation

To install *PComm*, please run **\Setup.exe** in the diskette.

Note that the MOXA Windows NT/95 device driver is required as using *PComm* diagnostic and monitor utilities.

5.2 Programming Library

The serial communication library is to assist users to develop programs for serial communications **for any COM port** complying with Microsoft Win32 API. It can ease the implementation of multi-process and multi-thread serial communication programs and hence greatly reduce the developing time.

For complete library function description and example programs for Visual C++, Visual Basic, Borland C++ and Delphi, please see help file and example programs in *PComm* directory for more details.

5.3 Utilities

The followings are short descriptions of each utility. For details, please see on-line help in the diskette.

Diagnostic

A convenient diagnostic program provides internal and external testing, such as IRQ, TxD/RxD, UART, CTS/RTS, DTR/DSR, DTR/DCD testing, etc., for the MOXA boards and ports to verify correct operation of both the software and hardware.

Monitor (Windows NT Only)

A useful port status monitoring program allows users to watch the selected MOXA COM ports' data transmitting/receiving throughput and communication line status which are updated and displayed on the screen at every time interval. In addition, users may click on one of the specific displayed port in order to see the current communication parameters and status of that port.

Chapter 6 For UNIX Users

There are various UNIX operating systems, such as SCO UNIX/XENIX, Solaris, AT&T UNIX SVR4.2, etc. Different types of UNIX drivers are required for different UNIXs. Moxa supports device drivers for most of the popular PC-based UNIXs mentioned above. Some obsolete UNIXs are no longer supported like Interactive UNIX, AT&T UNIX SVR3.x/4.0/4.1, etc.

In this chapter, driver installation procedure is described. Administration utility, **moxaadm**, is explained, which is good for configuration, monitor and terminal emulation. Related issues such as device naming, baud rate settings and terminal enable are stated. Additional `Ioctl()` commands are provided for advanced programming. Finally, UNIX-specific troubleshooting is included.

6.1 Driver Installation

At most 4 boards of combination of C320 *Turbo* and C218 Series can be installed together as long as memory resources are sufficient and available in a system. These boards share only one same IRQ.

Step 1: Boot the system and login from the console as a super user (root).

Step 2: Insert driver diskette into a 1.44 MB floppy drive, A: or B:.

If driver files are obtained from MOXA **FTP** service, put them under **/tmp/moxa** directory and **skip step 4**, the '**tar**' command.

Step 3: Change to root directory by entering the following command:

```
# cd /
```

Step 4: Enter the following command to extract the install script file first.

```
# tar xvf /dev/fd0135ds18 /tmp/moxa/install (if 1.44MB A:)
```

Step 5: Run installation command.

```
# /tmp/moxa/install
```

Now follow the prompted instructions to finish the driver installation.

Choose the listed and desired one when asked. Currently, MOXA supports SCO UNIX (and SCO OpenServer, SCO Open Desktop), UNIX SVR4.2 (and UnixWare), Solaris x86, and SCO XENIX. Ask your dealer for newly supported device drivers. If your system is none of the listed, choose the closest one to try.

Copyright (C) 199x Moxa Technologies Co., Ltd. All Rights Reserved.

MOXA UNIX Device Driver Installation Ver. x.x

Please select one of the following OSs:

1. SCO UNIX
2. UNIX SVR4.2
3. Solaris x86
4. SCO XENIX

Select :

Please select one of the devices where the driver diskette/files put:

- | | |
|--------------------|-------------|
| 1. /dev/fd0135ds18 | (A: 1.44MB) |
| 2. /dev/fd096ds15 | (A: 1.2MB) |
| 3. /dev/fd1135ds18 | (B: 1.44MB) |
| 4. /dev/fd196ds15 | (B: 1.2MB) |
| 5. Hard Disk | /tmp/moxa |

Select :

Then the **MOXA Multiport Board Installation Utility** will show for board installation.

Choose **C218/C320 Series driver** to be installed, instead of C168 Family. Please follow the instructions to install.

Step 6: Now you may shutdown the system and reboot. If you have trouble in booting the new UNIX kernel, use the backup kernel (**/unix.moxa** or

/xenix.moxa) to boot your system. The system will work as before. Refer to Appendix A for Troubleshooting.

For Solaris, boot the system by entering '**b -r**' at the boot prompt.

After the system starts up again, enter into the multiuser mode. The following message will show.

```
MOXA Serial I/O Board ( ver x.x )  
Downloading  
MOXA-C320 Series (XX000) is found.
```

The Basic Module LED display will stay at the last accessible port, e.g. "28" means that there are 2 UART Module/8-port units installed (if continuous 8 ports are regarded as an 8-port unit).

If the following message appears

```
WARNING! MOXA C320 Series (XX000) not found!
```

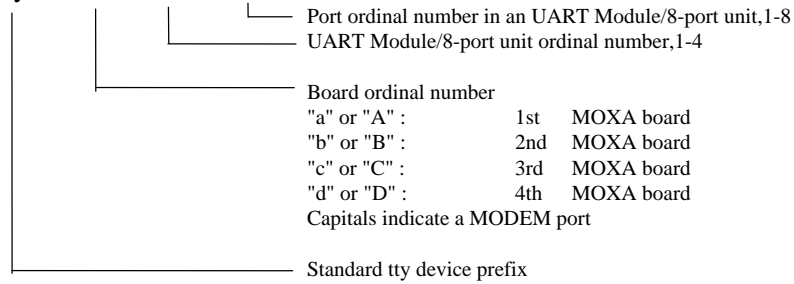
Please see Appendix "Troubleshooting".

6.2 MOXA TTY Device Names

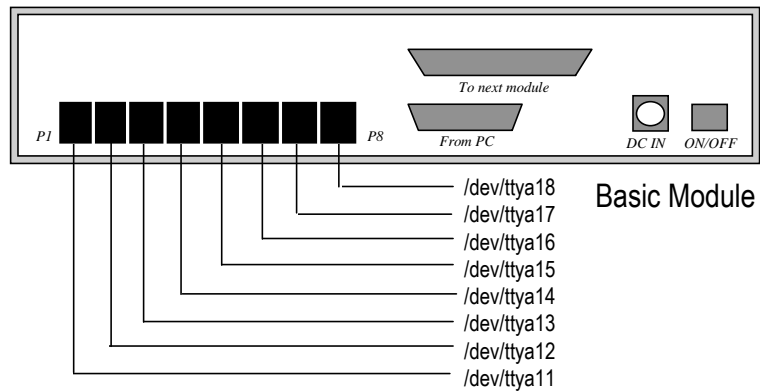
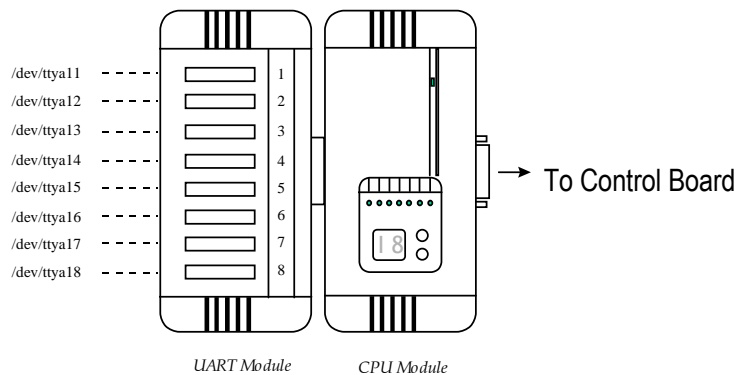
If the C320 *Turbo* is successfully configured, there will be two tty devices created for each port at **/dev** directory: one is **non-MODEM tty** (e.g. **ttya11**), and the other is **MODEM tty** (e.g. **ttyA11**). The two devices are actually accessing the same physical port except that the MODEM tty has to check the ON status of DCD signal to be able to open device, and closing device automatically as DCD signal is OFF.

The convention of the MOXA tty device name is **/dev/tty{a-d}{1-4}{1-8}**, where (if continuous 8 ports are viewed as an 8-port unit for C320 *Turbo* Rackmount):

"tty" + "A" + "B" + "C"



Taking 8 port C320 *Turbo* as an example,

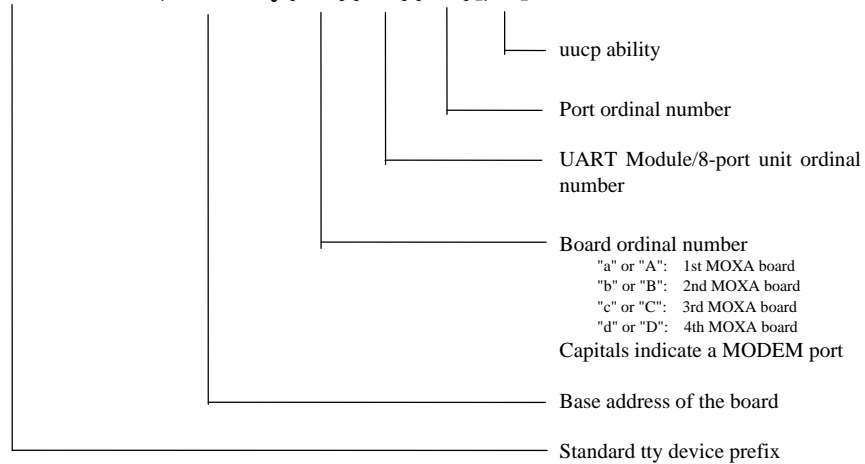


The Solaris device naming convention is different from other UNIXs which will be

explained below. For example, `/devices/isa/moxa@c8000:ttya12` indicates the second port on the first module (or unit) of the first MOXA board ('a') whose base address is C8000.

The convention of the Solaris tty device name is:

`/devices/isa/moxa@,xxxx:tty{a-d}{1-4}{1-8}[,cu]`



Another two device files will be created and linked to the corresponding device file `/devices/isa/moxa*` and `/devices/isa/moxa*,cu`, after system reboot with reconfiguration option. They are

```
/dev/term/number (linked to /devices/isa/moxa*)
/dev/cua/number  (linked to /devices/isa/moxa*,cu)
```

However, in order to comply with the Moxa device naming convention of other UNIXs, device files `/dev/tty{a-d}{1-4}{1-8}` and `/dev/tty{A-D}{1-4}{1-8}` are also created and linked to the corresponding Solaris device file `/devices/isa/moxa*` and `/devices/isa/moxa*,cu`. It is always helpful to know the relationship among the three device files:

```
/dev/tty{a-d}*  â /dev/term/number  â /devices/isa/moxa*
/dev/tty{A-D}*  â /dev/cua/number   â /devices/isa/moxa*,cu
```

6.3 Baud Rate Settings

Note also that the 50 baud rate, **B50**, will no longer stand for 50 bps, instead, it means 57600 bps, and **B75** for 115.2K bps, and **B110** for 230.4K bps and **B134** for 460.8K bps.

6.4 Administrations

Users can use the administration utility, **moxaadm**, to change the C320 *Turbo*'s basic and advanced configuration, to reset the board, to monitor the ports' activity, to tune the port performance, and to remove the installed MOXA driver from the UNIX/XENIX system.

Basic Configuration

Users can select board type, memory base, and interrupt as long as no conflict occurs. In main board settings dialog, **kernel rebuilding** is required if interrupt is changed. **Board reset** (described later) is available if any other setting is changed.

In [Basic Port Setting] sub-dialog, there are some noticeable fields:

RTS/CTS Hardware Flow Control:

This feature is for more flexible driver behavior. If set to OFF (default), CTS signal is not needed for tty port to transfer data and RTS/CTS hardware flow control function in driver is disabled. On contrary, if set to ON, CTS signal is needed for tty port to transfer data and RTS/CTS hardware flow control function in driver is enabled.

Non-Modem (Modem) Baud Rate: (not applicable for "Transparent" port or Solaris x86)

This field stands for the initial baud rate symbol and hunt sequence for Non-Modem (Modem) tty and is simply for setting parameters of getty entries in system file `/etc/inittab` which could also be manually modified by system administrator. Its value comes from the UNIX/XENIX system "getty default" file, e.g., `/etc/gettydefs`. Modify this field to suit your need.

In some cases, you may need to modify the "getty default" file as well. For example, in some UNIXs, the "9600" symbol indicated 8-data-bit no-parity while in others it may stand for 7-data-bit, even-parity. So, please examine the "getty default" file carefully,

and make sure the terminal settings is the same. Otherwise, garbled data will be inevitable.

Advanced Configuration

This is to provide advanced settings for advanced users beyond basic settings described in the previous **Basic Configuration**. Similarly, in main board settings dialog, **board reset** (described later) is available if any setting is changed.

On-Board Line Discipline:

In main board settings, this advanced feature is set to "YES" by default. By "YES" of On-Board Line Discipline we mean that all the tty line discipline jobs, e.g. translating NL to CR, will be done on C320 *Turbo*, instead of doing them on host. Hence, this will reduce the host work load significantly.

Terminal and Transparent:

Once On-Board Line Discipline is set to "YES", [Terminal] port feature is the only choice. This is good for terminal application users.

However, if On-Board Line Discipline is set to "NO", port feature could be either [Transparent] or [Terminal]. In case of [Terminal], all the tty line discipline jobs will be done on host which surely is a burden to the host.

In case of [Transparent], no time-consuming tty line discipline is applied for the port except Xon/Xoff flow control, thus non-terminal applications such as data acquisition will benefit from it, i.e. gain better throughput and performance. The C320 *Turbo* firmware will do Xon/Xoff flow control if ixon/ixany or ixoff are set via ioctl() system call. In this case, the Non-Modem and Modem Baud Rate do not apply.

For Solaris, to use a serial port as [Transparent] port, users have to make sure the 'ttypmon' (tty monitor) service for that port is disabled. If not, you must disable the port service via 'admintool' command under OpenWindows or 'pmadm' command. Here, the 'admintool' command is recommended.

Response:

Three levels, Slow, Normal, and Fast, of port response time are designed for user's need. In general, ports with faster response time will consume more host CPU's resource

while slower response consume less resource. If the port is specified as fast response, the driver will apply interrupt driver method to speed up the response. The response time for those ports with normal or slow response can be further tuned by the response level in 'Performance Tuning' menu.

Board Reset

The board reset utility eliminates the need to shut down the UNIX system to re-initialize the C320 *Turbo* with new configuration. However, before the board reset is issued, user must make sure that

1. All enabled ports must be disabled.
2. All processes, including getty or tty monitor, related to the C320 *Turbo* are killed.

Performance Tuning

The utility let user tune the favored response time for those ports with Resp = Norm or Slow in [Port Feature] sub-menu of Advanced configuration. Ten response time levels (0-9) are available for tuning. The idea is that the faster response time will consume more host CPU's resource while slower response consume less resource. If the host CPU is powerfully enough or a quick response is highly demanded, choose level 9 would be suitable.

Port Monitoring

This utility gives the user a quick view about all the MOXA ports' activities. One can easily learn each port's total received/transmitted (Rx/Tx) character count since the time when the monitoring is started. Rx/Tx throughputs per second are also reported in interval basis (e.g. the last 5 seconds) and in average basis (since the time the monitoring is started). You can reset all ports' count by <HOME> key. <+> <-> (plus/minus) keys to change the displaying time interval. Press <ENTER> on the port, that cursor stay, to view the port's communication parameters, signal status, and input/output queue.

Terminal Emulation

This utility provides data sending and receiving ability of all tty ports, especially for MOXA ports. It is quite useful for testing simple application, for example, sending AT command to a modem connected to the port or used as a terminal for login purpose. Note that this is only a dumb terminal emulation without handling full screen operation. Besides, data scope function with pattern/file transfer is provided.

Driver Removal

If you want to remove the MOXA C320 *Turbo* device driver and return to your system configuration, just press ENTER in this function entry and answer "y" to confirm. Then the system will be rebuild. This may take some time. If you answer "n", no action will be taken.

6.5 Setting MOXA Ports to Terminal

To enable the "Terminal" port for login purpose, taking ttya11 as an example,

for SCO UNIX/OpenServer/Open Desktop/XENIX,

```
# enable /dev/ttya11
```

for Solaris x86,

1. Enter Open Windows environment
2. # /usr/bin/admintool
3. Use 'Serial Port Manager' to enable the port or alternatively use 'pmdm' command to enable the port.

for UNIX SVR4.2 and UnixWare,

1. Edit (e.g. use vi editor) the file /etc/initab.
2. Modify the tty entry from "ma11:23:off:/etc/getty ttya11 9600" to "ma11:23:respawn:/etc/getty ttya11 9600".
3. # init q

Or refer to your UNIX system manuals for how to activate a tty port.

6.6 Programming the MOXA Ports

The system services that apply to standard tty port can also be applied to MOXA port since MOXA port conforms to UNIX tty standard. For example: `open()`, `ioctl()`, `read()`, `write()`, `close()`, etc. Please refer to your UNIX Programmer's Reference manual.

However, these system services only provide a limited functions and thus may not satisfy the sophisticated programmers' need. In order to meet the user's special purpose, MOXA supports extended services through `ioctl()` command.

Here is the list of the functions.

- | | |
|-----------------------|--|
| 1. MIBUFED (= 0x401) | To get byte count in input buffer. |
| 2. MOBUFED (= 0x402) | To get byte count in output buffer. |
| 3. MTCRTS (= 0x403) | To control RTS output signal. |
| 4. MTCDTTR (= 0x404) | To control DTR output signal. |
| 5. MLOWATER (= 0x405) | To set output buffer low water level. |
| 6. MSTATUS (= 0x407) | To read modem line status (CTS/DSR/DCD). |
| 7. MHWFLOW (= 0x40e) | To enable/disable the hardware flow control. |

For details, please refer to Appendix "Extended UNIX `ioctl()` Commands".

6.7 Troubleshooting

This section is specific for UNIX. For general problems and solutions please see Appendix "Troubleshooting".

1. Under UNIX/XENIX, MOXA port responds too slowly or abnormal key operation.

Hardware Cause and Solution:

- Wrong IRQ setting. If the hardware and software IRQ setting do not match, MOXA driver will use a polling scheme and thus causes slow operation. Try to match the IRQ settings.

Software Cause and Solution:

- a. The applications are not compatible with the C320 *Turbo* On-board line discipline. The only way to work this around is to set On-board line discipline to "No" in [Board] sub-menu of Advanced configuration in **moxaadm**.
2. Under UNIX/XENIX, when respawning quite a few number of tty ports, the following error messages appear: "Time out table overflow", "File table overflow", "Region table overflow".

Software Cause and Solution:

The above error messages imply that the system resources are exhausted. You should tune the kernel parameters to a larger value and rebuild the kernel to be able to accommodate the new configuration. Refer to UNIX system manual about how to tune the parameters and rebuild kernel.

"Time out table overflow" : NCALL parameter too small.
 "File table overflow" : NFILE or NINODE parameter too small.
 "Region table overflow" : NREGION or NPROC parameter too small.

3. Under UNIXs, the newly rebuilt kernel could not boot.

Software Causes and Solutions:

The C320 *Turbo* driver might not be built into the new kernel correctly.

- a. Please use the last good kernel backup to boot again. The kernel backup in SCO UNIX is /unix.moxa while in UNIX SVR4.2 /stand/unix.moxa.
- b. Then remove the C320 *Turbo* driver. Refer to Section 6.4 for driver removal.
- c. Re-install the C320 *Turbo* driver once more.
4. Under UNIXs, when using MOXA port for PPP/SLIP connection, what should the On-Board Line Discipline feature be set to?.

Software Cause and Solution:

Set to OFF. Otherwise the PPP/SLIP connection will fail.

Chapter 7 For DOS Users

The C320 *Turbo* is useful for office automation and industrial control as well as data acquisition and communication. Because many users need to implement an industrial control system or data acquisition system in the DOS environment, but DOS did not have any better communication API, thus Moxa has developed device drivers and library functions under DOS. Moxa device driver is a port-based one and supports maximum of **256** serial ports. Furthermore, Data Scope utility is very helpful for troubleshooting and debugging the serial communications.

In this chapter, driver installation, setup, loading and unloading procedure are described. Utility, Data Scope, is explained, which is good for terminal emulation, data scope and diagnostics. Related issues such as device naming and programming are also stated.

7.1 Driver Installation, Setup, Loading and Unloading

MOXA DOS API-232 is a software package that assists users to develop and/or debug programs for serial communications. It supports device drivers for the MOXA boards such as C320 *Turbo*, C218 Series, C102/C104/C168 Family, standard PC COM ports, etc.

Driver Installation

Run the installation program, **DOSINST.EXE**, in the API-232 software diskette. Specify the target API-232 directory (e.g. **C:\MOXA**) where software driver will be copied to. Press **F2** to start the installation.

Driver Setup

Change to API-232 directory and run the setup program, **BIN\SETUP.EXE**, then choose "**C218/C320 Async Multiport Board**" in the "DRIVER SELECTION" dialog and then specify the board type, modules, port number range, base memory

address, and IRQ.

Type: Choose the board type, C320 *Turbo*.

Ports: The number of ports for C320 *Turbo*.

Port number: The port ID of each port. The application software will refer to the port by its port number (ID). Duplicated port number is not allowed.

Memory base address: The base address of each board. Duplication is not allowed.

Interrupt number: The IRQ number of the board. All C320 *Turbo* and C218 Series must share one IRQ only.

The maximum supported number of C320 *Turbo* and C218 Series boards is limited to 4 in one system. For example, you can install one C320 *Turbo* and three C218, or two C320 *Turbo* and two C218, or four C320 *Turbo* under the DOS system.

On the second dialog "PORT SETUP", enter/modify each port's configuration. These are the port initial values when driver is loaded. Some noticeable fields are explained below.

On board TxD buf : The transmit (output) buffer already allocated on the C320 *Turbo* or C218 Series board for each port. Cannot be altered.

On board RxD buf : The receive (input) buffer already allocated on the C320 *Turbo* or C218 Series board for each port. Cannot be altered.

External RxD buf : Extra receive (input) buffer you wish to allocate for each port. This will consume the DOS conventional memory. Default value is 0.

Driver Loading

Having completed the setup, load the driver, **BIN\MX-DRV.EXE**, at the DOS prompt. The driver will detect the C320 *Turbo* automatically. If all the Control Board(s) and external module(s) are detected, a message similar to below will show:

```
API-232 Version x.x
MOXA C218/C320 Series Communication Driver
Setup driver...
C320 Turbo multiport board 1 : bank [C800] port [1-8] IRQ 15!!
Device driver setup O.K.
```

, which means the C320 *Turbo* driver is installed properly. At this point, user is ready to execute the application written in API-232 functions, or start developing applications.

If the board does not match the setup data, the screen will show a message like:

```
API-232 Version x.x
MOXA C218/C320 Series Communication Driver
Setup driver...
C320 Turbo multiport board 1 : bank [C800] not found!!
Driver execution failure!
```

, which means the C320 *Turbo* driver is not installed properly. Please refer to Appendix "Troubleshooting" for possible reasons and solutions.

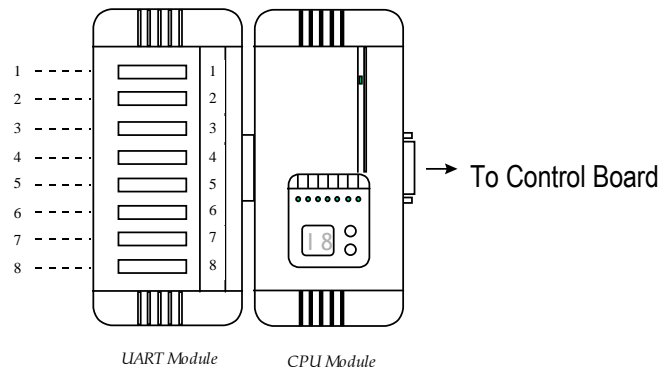
Driver Unloading

To unload the C320 *Turbo* driver from memory, type **MX-DRV/Q** at the DOS prompt.

7.2 MOXA Serial Port Names

Each MOXA serial port are referenced as **port number** in terms of programming. The port numbers are automatically assigned once the starting port number is decided by user when configuring the ports of the board.

For example, if 8 ports are configured and the starting port number is 1, then the mapping of serial port numbers will be as depicted.

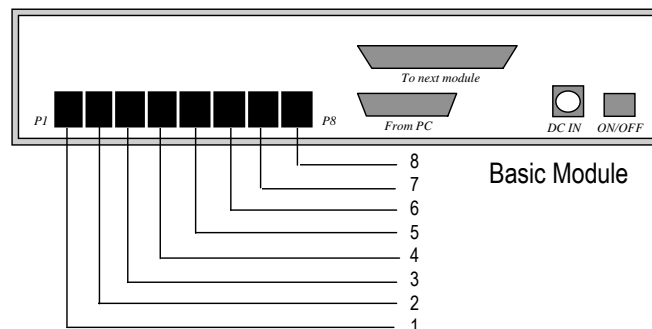


7.3 Programming Library

For DOS programming, MOXA API-232 includes powerful libraries supporting languages like Microsoft C, Turbo C, Macro Assembly, QuickBasic, Turbo Pascal, Clipper, etc. Sample programs for each supported language are included, and placed in the sub-directory `..\EXAMPLE\language` of the API-232 directory.

In addition, for DOS C language only, there are also Modem Control and File Transfer library available, supporting Hayes compatible modem control as well as ASCII, KERMIT, XMODEM, YMODEM, and ZMODEM file transfer protocol functions.

For complete API-232 function description, see file **API-232.TXT** in the API-232 directory for more details.



The following functions illustrated in C are for C320 *Turbo* only. Function names may vary in some language.

1. `sio_bank()`: Get C320 *Turbo* memory base address.
2. `sio_linked()`: Check if CPU/Basic Module of C320 *Turbo* is linked.

7.4 Data Scope

The Data Scope, **BIN\SCOPE.EXE**, is a suite of utility programs that can help users with system troubleshooting and serial communication debugging. There are three major functions:

1. The Data Scope utility offers transparent monitoring of serial communication lines and allows data to be streamed to disk storage for later analysis.
2. The TTY terminal emulation utility allows user to view the signal status and transfer data interactively or files using ASCII, XMODEM, YMODEM, ZMODEM and KERMIT protocols.
3. The diagnostic test utility provides port connection test with two MOXA ports connected via a properly-wired cable.

Please see on-line help as running BIN\SCOPE.EXE for more usage and capability information.

Chapter 8 For Windows 3.x Users

For users who need to implement an industrial control system or data acquisition system in the Windows 3.x environment, Moxa has developed two device drivers and/or library functions under Windows 3.x which is called Windows 3.x API-232. You should find the API-232 driver diskette along with shipment in addition to API-232 User's Manual.

In this chapter, driver installation and configuration procedure are described. Utility, **TTY**, is explained, which is good for terminal emulation. Related issues such as driver removal, programming and existing applications are also stated.

8.1 Driver Installation

MOXA Windows 3.x API-232 is a software package that assists users to develop and/or debug programs for serial communications. It includes two different types of Windows 3.x drivers:

1. MOXA Windows-compatible COMM Driver: for the MOXA C104 Family, C218/C320 Series only.
2. MOXA Proprietary Windows DLL Driver: for C218/C320 Series only.

The installation program, **WININST.EXE**, which asks the user to choose the desired driver for the selected MOXA board and also prevents the possibility of driver installation confusion. The two types of drivers could not coexist in a system, in other word, only one type of driver could be enabled in one system and removal of previous different type of driver is required.

Carefully examine which one of the two drivers is suitable for you. Here is our suggestion: If you need no more than 9 serial ports or intend to run an already made software program (such as pcANYWHERE for Windows) that supports Windows Win16 SDK COMM API calls, then choose MOXA Windows-compatible COMM Driver as your platform. If you need more than 9 serial ports or have been familiar with the MOXA API-232 functions, then MOXA Proprietary DLL Driver would be suitable for you. In this case, developing the Windows application by yourself is necessary.

8.2 MOXA Windows-compatible COMM Driver

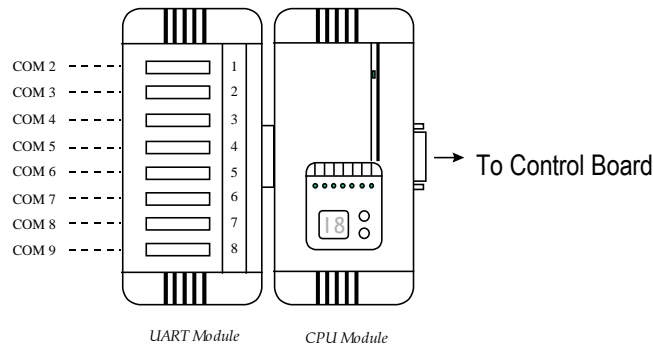
Due to the limitations of Windows 3.x operating system itself, only up to 9 COM ports are supported, i.e., COM1 to COM9. Hence, C320 with 8 ports is a suitable choice.

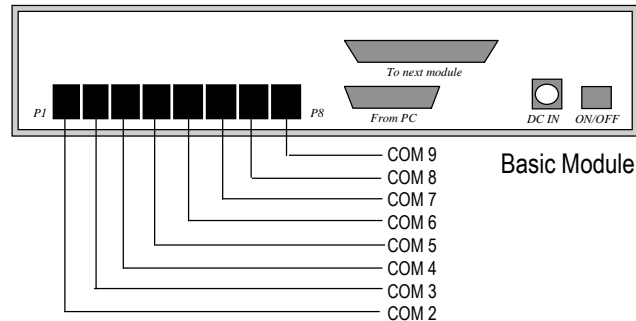
After successful installation, a program group will be created which contains all the useful programs Moxa provided. Restarting of Windows 3.1 system is required for MOXA Standard Windows-compatible COMM Driver after installation. When system restarted, the MOXA COMM Driver is ready to go.

Configuration

The Windows-based configuration program, **CONFIG**, has the easiest way to configure the 8 ports of C320 *Turbo*, either from COM1 to COM8, from COM2 to COM9, from COM3 to COM9, from COM4 to COM9 or from COM5 to COM9, depending on user's need. Normally COM1 is used by mouse and to fully use the MOXA 8 ports, thus COM2 to COM9 is recommended. In this case, the original COM2 on PC will be ignored, no matter it is disabled or not and any thing connecting to COM2 could now connect to the first port of C320 *Turbo*, instead. In case of configuring from COM1 to COM8, the situation is similar to the previous one and the first port of C320 *Turbo* serves as original COM1, while the second port as COM2. As to configuring from COM3 to COM9, only 7 ports are workable, and so on.

For example, if 8 ports are configured and starting from COM2, then the mapping of serial port numbers will be as depicted.





Utilities

The utility, **TTY**, is included to help users monitor and debug RS-232 communications under Windows 3.x which can manipulate ports from COM1 to COM9. It is just a simplest example program which can send and receive data after each port opened with selected communication parameters. Multiple windows for ports simultaneously are available for a demonstration of multitasking feature of Windows 3.x. The Windows-provided application, **Terminal**, can only make use of COM1 to COM4 which is obviously a restriction.

Windows 3.x COMM API Programming

MOXA Windows-compatible COMM Driver supports **Microsoft Windows COMM API** such as OpenComm(), ReadComm(), WriteComm(), etc. It supports any language conforming to the Windows COMM API like Microsoft C, Borland C, Visual C, Visual Basic, Delphi, etc. Sample programs for only Microsoft C, Borland C and Visual Basic are supported. For other languages' sample programs, please refer to the language-provided communication example programs.

Application Note

Many Windows software packages, such as pcANYWHERE, LabView, FIX, WinFax Pro, Fax Server, PROCOMM PLUS, etc., can access MOXA COM ports directly since these applications follow the Microsoft Windows COMM API. On the contrary, those Windows applications not fully compatible with

Windows COMM API standard might not be able to operate MOXA COM ports. For instance, LapLink for Windows accesses I/O address directly which will fail with the C320 *Turbo* because C320 *Turbo* is accessed via memory instead of I/O port.

8.3 MOXA Proprietary Windows DLL Driver

For MOXA Proprietary DLL Driver, MOXA API-232 library functions are supported for programming, instead of Windows COMM API. The best advantage of using MOXA Proprietary DLL Driver is that users can have up to 128 ports in a Windows system which break the restriction of 9 COM ports.

After successful installation, a program group will be created which contains all the useful programs Moxa provided. Restarting of Windows 3.x system is not necessary for MOXA Proprietary DLL Driver after installation. It is the dynamic linkage feature that enable the driver loaded dynamically.

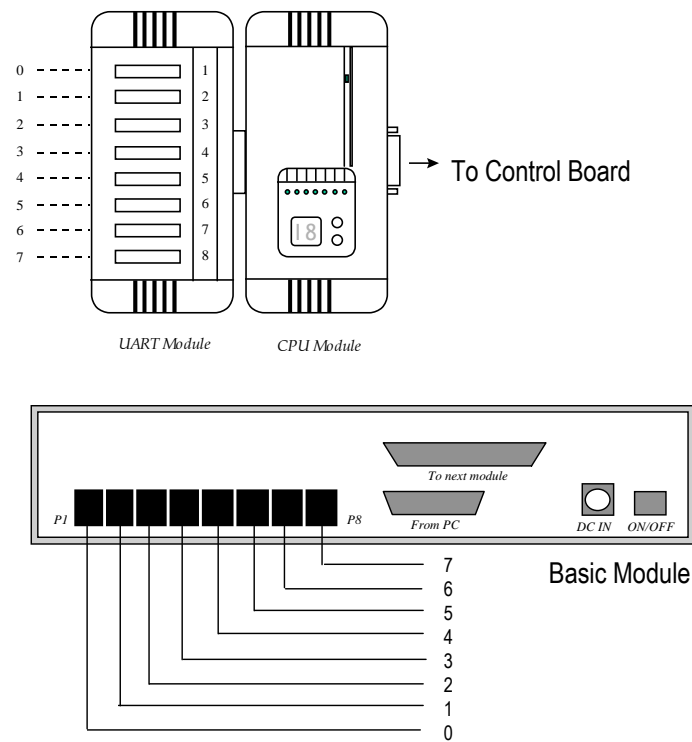
Configuration

The Windows-based configuration program, **CONFIG**, has the easiest way to configure up to **128** ports in one system. Max. of four C320 *Turbo* are allowed to be installed in one system and it makes 128 ports in total. The port number is simply used to identify the port. Never confuse the port number with COM1 to COM9 which are totally different things. The starting port number is freely selectable and the last port number configured will not exceed 128.

For example, if 8 ports are configured and the starting port number is 0, then the mapping of serial port numbers will be as depicted.

Utilities

The utility, **TTY**, is included to help users monitor and debug RS-232 communications under Windows 3.x which can manipulate port from number 0 to 127. It is just a simplest example program which can send and receive data after each port opened with selected communication parameters. Multiple windows for ports simultaneously are available for a demonstration of multitasking feature of Windows 3.x. The Windows-provided application, **Terminal**, can not be used in this case since the MOXA Proprietary DLL Driver is not used for applications written in Microsoft Windows COMM API.



MOXA Windows API-232 Programming

MOXA API-232 Library is available for programming under Microsoft Windows 3.x as MOXA Proprietary Windows DLL Driver is installed. API-232 functions, such as

sio_open(), sio_read() and sio_write(), could be used in a way that is mostly like DOS API-232 programming. It supports languages under Windows like Microsoft C, Borland C, Visual C, Visual Basic, Delphi, etc. Sample programs for Visual C and Visual Basic are supported in the driver file directory after driver installation.

In addition, for Windows C language only, there are also Modem Control and File Transfer library available, supporting Hayes compatible modem control as well as ASCII, KERMIT, XMODEM, YMODEM, and ZMODEM file transfer protocol functions.

For complete Windows 3.x API-232 function description, see **API-232 User's Manual** or file **API-232.TXT** in the API-232 library for more details.

The following functions are for Windows 3.x only.

1. sio_init(): MOXA Proprietary Windows DLL Driver initialization.
2. sio_end(): Release the MOXA Windows Proprietary DLL driver.

Appendix A Troubleshooting

Common C320 *Turbo* problems and possible solutions are listed below. If you still have problems, contact your dealer or Moxa for help. Or report it using "**Problem Report Form**" to your dealer at once for technical support.

1. MOXA board not found.

Hardware causes and solutions:

- a. The base address you have selected is conflicting with the BIOS shadow RAM or cache. Please disable the area for board.

The base address you have selected for board conflicts with that of other interface boards such as SCSI, LAN or Sound boards. If so, adjust the address to forestall the conflict.

The base address you have selected for the board conflicts with expansion memory. If so, adjust the address to avoid the conflict.

You can use the F4 "Find free Bank" function of DOS utility "SETUP" in the API-232 to help find the system's free bank.

- b. The board is not properly plugged in the system. If that is the case, then re-install the board and make sure that it fits well in a 16-bit slot this time. Sometimes the slot for plugging board is bad. In this case, please try other slots until a good one is found.
- c. The board is not installed or missing (absent). Please install it.
- d. The board is out of order. Please check the board(s) ONE AT A TIME in the system and find out the defective one. It is also recommended to check it under DOS with F3 Diagnostics function of utility "SETUP" to verify the hardware conditions.
- e. Sometimes it happens when the total consumption of power in the system is higher than the power supply can provide. Especially when many

interface boards including board(s) are plugged in the system. It is recommended to replace the current power supply with higher watt one. Or choose the external power option if C320 *Turbo* is used.

Software Causes and Solutions:

- a. Under DOS/Windows 3.x system, you may have selected a base address that conflicts with other driver such as EMM386 or QEMM386 memory management software. If so, reserve 16K bytes space for C320 *Turbo*, e.g. using "X = C800-CBFF" option in EMM386/QEMM386 command line of file CONFIG.SYS.
- b. The base address switch setting on board is not the same as the setting in configuration software.

2. MOXA board found but can not transfer data.

Hardware Causes and Solutions:

- a. Check if wrong cable wiring. Refer to Appendix B.3 for precise pin assignment of the connector type you are using.
- b. Cable or external module is defective.
- c. IRQ is used or reserved by PCI bus.

Software Causes and Solutions:

- a. C320 *Turbo* will always check line status (CTS/DSR) before it sends data out if RTS/CTS flow control feature is set to ON in configuration or application program. Please see Appendix B.3.1/B.3.2, Type 3 for proper wiring.

Check the lines of the suspect port using the diagnostic LED indicators on the Basic Module if C320 *Turbo* is used.

- b. Perhaps the application controlling MOXA board is not correctly written according to the corresponding API of the operating system. To verify, please run existing and known good application or Moxa-provided utility. For example, under Windows 95, "Hyperterminal" is good for testing COM ports. Under UNIX, "stty", "cat", or "getty" is existing program for testing tty port.

- c. IRQ settings of software and hardware do not match.

3. "CPU/Basic Module not found" message.

Or CPU/Basic Module can not download firmware from the Control Board.

Hardware Causes and Solutions:

- a. Power off the CPU/Basic Module and Control Board system.
Check that the connection cable between the CPU/Basic Module and the Control Board is correct and firmly plugged in on both sides.
Power on the CPU/Basic Module and wait until the LED display reads "Ld". Power on the Control Board system.
- b. Sometimes it happens when the total consumption of power in the system is higher than the power adapter can provide. Especially when many interface boards including MOXA board(s) are plugged in the system. It is recommended to replace the current power adapter with higher watt one. Or choose the external power option if C320 *Turbo* is used.
- c. If all the possible methods fail, the CPU/Basic Modules might be defective. It is recommended to check it under DOS with utility "SETUP" to verify the hardware conditions.

4. Error messages on CPU Module LED display.

Any of the following messages means failure of the C320 *Turbo* initialization.

- E0 : CPU Module ROM error.
- E1 : CPU Module RAM error.
- E2 : No UART Module present.
- A1 : CPU Module execution failure.
- A2 : CPU Module can not communicate with Control Board.
- 10 : The first UART Module initialization failure.
- 20 : The second UART Module initialization failure.
- 30 : The third UART Module initialization failure.
- 40 : The forth UART Module initialization failure.

Hardware Causes and Solutions:

If the error message persists after check, return the unit for repair.

As "E0", "E1" or "A1" occurs, usually users have to power off and on the CPU Module. If problem persists, the CPU Module might be defective.

As "E2" occurs, turn off the CPU Module, firmly connect the UART Modules, and turn the power back on.

As "A2" occurs, check the link between CPU Module and Control Board is fastened carefully.

As "10", "20", "30" or "40" occurs, make sure the connection between modules is proper.

5. Error messages on Basic Module LED display.

Any of the following messages will halt the Basic Module.

E4 The power to the Basic Module has been switched off and on swiftly, or some Module is missing during the power-off time.

E5 The connection between Modules is loose or broken.

Hardware Causes and Solutions:

As "E4" occurs, usually users have to power on the Basic Module. Or check if the total number of ports (or modules) is correct or not.

As "E5" occurs, it may indicate that the cables between modules should be fastened carefully.

6. Upon system startup or driver initialization, the following message appears: "Number of ports mismatch".**Hardware Causes and Solutions:**

The total number of ports installed is less than that of configured.

- a. Shut off the system and check all modules and their connections. Make sure that all the module(s) is(are) fastened properly.

- b. Run the board configuration program to check the board configuration.
For C320 *Turbo*, connect the CPU/Basic Module with proper number of UART/Extensive Module(s) as configured in configuration program.
- c. If all the possible methods fail, one of the module(s) might be defective.
Please check the module(s) ONE AT A TIME in the system to find out the defective one. It is also recommended to check it under DOS with utility "SETUP" to verify the hardware conditions.

Appendix B Technical Reference

B.1 Dual-Ported RAM

There is 512 KB of dual-ported DRAM on the C320 *Turbo* Control Board which is used as a data buffer and memory map device for the motherboard. It is called "dual-ported" because it is accessed by both motherboard CPU and the C320 *Turbo* Control Board's on-board CPU. Generally speaking, the memory access time is much faster than normal I/O access.

All messages and data are transferred through this 512 KB dual-ported memory. The buffer area appears as normal 16-bit memory to the PC AT I/O channel.

Every C320 *Turbo* Control Board only occupies 16 KB of memory space (window) from the system point of view. There are six possible base addresses you can choose from, which are below 1 MB. Please see Chapter 2/3 for dip switch settings. For UNIX users, 36 more base addresses are available, which are above 1 MB and below 16 MB. The followings are the SW1 dip switch settings for base addresses above 1 MB and below 16 MB. However, the more RAM installed in the PC, the less base addresses available under UNIX.

SW1 Dip switch settings

1	2	3	4	5	6
ON	OFF	ON	OFF	ON	ON
OFF	OFF	ON	OFF	ON	ON
ON	ON	OFF	OFF	ON	ON
OFF	ON	OFF	OFF	ON	ON
ON	OFF	OFF	OFF	ON	ON
OFF	OFF	OFF	OFF	ON	ON
ON	OFF	ON	ON	OFF	ON
OFF	OFF	ON	ON	OFF	ON
ON	ON	OFF	ON	OFF	ON
OFF	ON	OFF	ON	OFF	ON
ON	OFF	OFF	ON	OFF	ON
OFF	OFF	OFF	ON	OFF	ON
ON	OFF	ON	OFF	OFF	ON

Base Address
4C8000
4CC000
4D0000
4D4000
4D8000
4DC000
8C8000
8CC000
8D0000
8D4000
8D8000
8DC000
AC8000

OFF	OFF	ON	OFF	OFF	ON
ON	ON	OFF	OFF	OFF	ON
OFF	ON	OFF	OFF	OFF	ON
ON	OFF	OFF	OFF	OFF	ON
OFF	OFF	OFF	OFF	OFF	ON
ON	OFF	ON	ON	ON	OFF
OFF	OFF	ON	ON	ON	OFF
ON	ON	OFF	ON	ON	OFF
OFF	ON	OFF	ON	ON	OFF
ON	OFF	OFF	ON	ON	OFF
OFF	OFF	OFF	ON	ON	OFF
ON	OFF	ON	OFF	ON	OFF
OFF	OFF	ON	OFF	ON	OFF
ON	ON	OFF	OFF	ON	OFF
OFF	ON	OFF	OFF	ON	OFF
ON	OFF	OFF	OFF	ON	OFF
OFF	OFF	OFF	OFF	ON	OFF
ON	OFF	ON	ON	OFF	OFF
OFF	OFF	ON	ON	OFF	OFF
ON	ON	OFF	ON	OFF	OFF
OFF	ON	OFF	ON	OFF	OFF
ON	OFF	OFF	ON	OFF	OFF
OFF	OFF	OFF	ON	OFF	OFF

ACC000
AD0000
AD4000
AD8000
ADC000
CC8000
CCC000
CD0000
CD4000
CD8000
CDC000
DC8000
DCC000
DD0000
DD4000
DD8000
DDC000
FC8000
FCC000
FD0000
FD4000
FD8000
FDC000

B.2 TI550C in UART/Basic/Extensive Modules

The TI550C, compatible with 16550C is an advanced asynchronous controller capable of on-chip hardware flow control and highest speed up to 921.6 Kbps. Inside UART/Basic/Extensive Module, each TI550C chip control one port.

B.3 Cable Wiring

In data communications, the term **DTE** is Data Terminal Equipment like terminal or PC COM1/2. The term **DCE** is Data Communication Equipment like modem. Please check the precise pinouts, the following pinouts are typical examples.

The following table shows EIA Standards RS-232 and RS-422, for reference only.

	Mode of operation	No. of Drv.	No. of Rcv.	Max. length	Max. Rate (bps)
RS-232	single ended	1	1	15 m (50 ft)	9600
RS-422	differential	1	10	1.2Km (4000 ft)	10M

B.3.1 RS-232/422 Cable Wiring for UART Modules (Desktop Option)

RS-232 Cable Wiring

The following RS-232 UART Modules are designed for C320 *Turbo*.

C32045T : UART Module providing 8 female RS-232 ports.

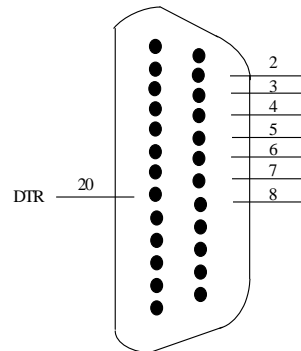
C32047T : UART Module providing 8 male RS-232 ports.

C32071T : UART Module providing 8 female RS-232 ports with 2000V surge protection which can protect ports from lightning.

There are two different types of pinout for RS-232 UART Modules. One is the pinouts for **female** ports, the other is for **male** ports. Identify the type of the UART Module and choose the correct cable wiring.

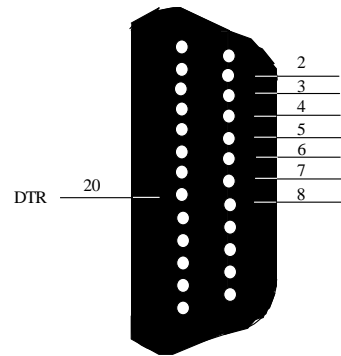
UART Module C32045T/71T
DB25 Female Connector

3	TxD
2	RxD
5	RTS
4	CTS
20	DSR
7	GND
8	DCD
6	DTR



UART Module C32047T
DB25 Male Connector

2	TxD
3	RxD
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
20	DTR



Type 1: To connect UART Module to a DTE.

UART Module DB25 Female			DTE Device DB25 Male			UART Module DB25 Female			DTE Device DB9 Male		
TxD	3	——	3	RxD		TxD	3	——	2	RxD	
RxD	2	——	2	TxD		RxD	2	——	3	TxD	
RTS	5	——	5	CTS		RTS	5	——	8	CTS	
CTS	4	——	4	RTS		CTS	4	——	7	RTS	
DTR	6	——	6	DSR		DTR	6	——	6	DSR	
DSR	20	——	20	DTR		DSR	20	——	4	DTR	
GND	7	——	7	GND		GND	7	——	5	GND	
DCD	8	——	8	DCD		DCD	8	——	1	DCD	

UART Module DB25 Male			DTE Device DB25 Male			UART Module DB25 Male			DTE Device DB9 Male		
TxD	2	——	3	RxD		TxD	2	——	2	RxD	
RxD	3	——	2	TxD		RxD	3	——	3	TxD	
RTS	4	——	5	CTS		RTS	4	——	8	CTS	
CTS	5	——	4	RTS		CTS	5	——	7	RTS	
DTR	20	——	6	DSR		DTR	20	——	6	DSR	
DSR	6	——	20	DTR		DSR	6	——	4	DTR	
GND	7	——	7	GND		GND	7	——	5	GND	
DCD	8	——	8	DCD		DCD	8	——	1	DCD	

Type 2: To connect UART Module to a DCE.

UART Module DB25 Female			DCE Device DB25 Female			UART Module DB25 Male			DCE Device DB25 Female		
TxD	3	——	2	RxD		TxD	2	——	2	RxD	
RxD	2	——	3	TxD		RxD	3	——	3	TxD	
RTS	5	——	4	CTS		RTS	4	——	4	CTS	
CTS	4	——	5	RTS		CTS	5	——	5	RTS	
DTR	6	——	20	DSR		DTR	20	——	20	DSR	
DSR	20	——	6	DTR		DSR	6	——	6	DTR	
GND	7	——	7	GND		GND	7	——	7	GND	
DCD	8	——	8	DCD		DCD	8	——	8	DCD	

Type 3: To connect UART Module to a DTE with 3-pin wiring.

If [Hardware flow control] feature is set to "ON", user must loop back (or short) the RTS with CTS and DSR with DTR, DCD on MOXA site.

UART Module DB25 Female			DTE Device DB25 Male			UART Module DB25 Female			DTE Device DB9 Male		
TxD	3	——	3	RxD		TxD	3	——	2	RxD	
RxD	2	——	2	TxD		RxD	2	——	3	TxD	
GND	7	——	7	GND		GND	7	——	5	GND	
RTS	5	┌	5	CTS		RTS	5	┌	8	CTS	
CTS	4	└	4	RTS		CTS	4	└	7	RTS	
DTR	6	┌	6	DSR		DTR	6	┌	6	DSR	
DSR	20	└	20	DTR		DSR	20	└	4	DTR	
DCD	8	┌	8	DCD		DCD	8	└	1	DCD	

UART Module DB25 Male			DTE Device DB25 Male			UART Module DB25 Male			DTE Device DB9 Male		
TxD	2	——	3	RxD		TxD	2	——	2	RxD	
RxD	3	——	2	TxD		RxD	3	——	3	TxD	
GND	7	——	7	GND		GND	7	——	5	GND	
RTS	4	┌	5	CTS		RTS	4	┌	8	CTS	
CTS	5	└	4	RTS		CTS	5	└	7	RTS	
DTR	20	┌	6	DSR		DTR	20	┌	6	DSR	
DSR	6	└	20	DTR		DSR	6	└	4	DTR	
DCD	8	┌	8	DCD		DCD	8	└	1	DCD	

If [Hardware flow control] feature is set to "OFF", users could just leave RTS, CTS, DSR, DTR, DCD open.

UART Module DB25 Female			DTE Device DB25 Male			UART Module DB25 Female			DTE Device DB9 Male		
TxD	3	——	3	RxD		TxD	3	——	2	RxD	
RxD	2	——	2	TxD		RxD	2	——	3	TxD	
GND	7	——	7	GND		GND	7	——	5	GND	
RTS	5		5	CTS		RTS	5		8	CTS	
CTS	4		4	RTS		CTS	4		7	RTS	
DTR	6		6	DSR		DTR	6		6	DSR	
DSR	20		20	DTR		DSR	20		4	DTR	
DCD	8		8	DCD		DCD	8		1	DCD	

UART Module DB25 Male			DTE Device DB25 Male			UART Module DB25 Male			DTE Device DB9 Male		
TxD	2	——	3	RxD		TxD	2	——	2	RxD	
RxD	3	——	2	TxD		RxD	3	——	3	TxD	
GND	7	——	7	GND		GND	7	——	5	GND	
RTS	4		5	CTS		RTS	4		8	CTS	
CTS	5		4	RTS		CTS	5		7	RTS	
DTR	20		6	DSR		DTR	20		6	DSR	
DSR	6		20	DTR		DSR	6		4	DTR	
DCD	8		8	DCD		DCD	8		1	DCD	

RS-422 Cable Wiring

The following RS-422 UART Modules are designed for C320 *Turbo*.

- C32061T:** UART Module providing 8 female RS-422 DB25 ports without isolation protection.
- C32065T:** UART Module providing 8 female RS-422 DB25 ports with 500V isolation protection which can prevent damage caused by high potential voltage.

The RS-422 transmission distance can reach as long as 1.2Km (4000ft) at 10Mbps.

Pinouts and RS-422 Cable Wiring

C32061T/65T		Remote site
3	TxD+(B)	RxD+(B)
16	TxD-(A)	RxD-(A)
2	RxD-(B)	TxD+(B)
14	RxD-(A)	TxD-(A)
7	GND	GND
5	RTS+(B)	CTS+(B)
13	RTS-(A)	CTS-(A)
4	CTS+(B)	RTS+(B)
19	CTS-(A)	RTS-(A)

Impedance Matching

When an electrical signal travels through two different resistance junctions in a transmission line, the mismatch will sometimes cause signal reflection. Signal reflection causes signal distortion, which in turn will contribute communication errors. The solution to this problem is to establish the same impedance at the line ends as in the line itself by terminating them with resistors.

It is normally sufficient when the value of the termination resistor equals the characteristic impedance of the transmission line. The resistors should be added near the receiving side. See below.

C32061T/65T		Remote site
3	TxD+(B)	RxD+(B)
16	TxD-(A)	RxD-(A)
2	RxD+(B)	TxD-(B)
14	RxD-(A)	TxD-(A)
5	RTS+(B)	CTS+(B)
13	RTS-(A)	CTS-(A)
4	CTS+(B)	RTS+(B)
19	CTS-(A)	RTS-(A)

Note:

1. ξ stands for termination resistor near the receiving side.
2. The suggested termination resistor for AWG #26 cable is 100 ohm.
3. The suggested termination resistor for phone cable is 600 ohm.

B.3.2 RS-232 Cable Wiring for Basic/Extensive Module (Rackmount Option)

For Basic/Extensive Module, only RS-232 interface with RJ-45 female connector is supported. A 1.5-meter male RJ-45 to male DB25 RS-232 cable is provided with shipment.

The following RS-232 Basic Modules are designed for C320 *Turbo*.

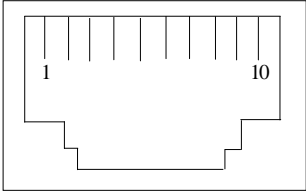
- C32080T :** Basic Module providing 8 female RS-232 ports.
C32081T : Basic Module providing 16 female RS-232 ports.

The following RS-232 Extensive Modules are designed for C320 *Turbo*.

- C32082T :** Extensive Module providing 8 female RS-232 ports.
C32083T : Extensive Module providing 16 female RS-232 ports.

C32080T/81T/82T/83T RJ-45 Jack

Pin	Signals
1	DCD
2	DSR
3	RTS
4	GND
5	TxD
6	RxD
7	GND
8	CTS
9	DTR
10	-



RJ45 JACK

Type 1: To connect Basic/Extensive Module to a DTE.

C320 <i>Turbo</i> RJ-45 Jack		DTE Device DB25 Male		C320 <i>Turbo</i> RJ-45 Jack		DTE Device DB9 Male	
DCD	1	8	DCD	DCD	1	1	DCD
DSR	2	20	DTR	DSR	2	4	DTR
RTS	3	5	CTS	RTS	3	8	CTS
GND	4/7	7	GND	GND	4/7	5	GND
TxD	5	3	RxD	TxD	5	2	RxD
RxD	6	2	TxD	RxD	6	3	TxD
CTS	8	4	RTS	CTS	7	7	RTS

DTR	9	6	DSR	DTR	8	6	DSR
-----	---	---	-----	-----	---	---	-----

Type 2: To connect Basic/Extensive Module to a DCE.

C320 Turbo RJ-45 Jack				DCE Device DB25 Female			
DCD	1	————		8	DCD		
DSR	2	————		6	DTR		
RTS	3	————		4	CTS		
GND	4/7	————		7	GND		
TxD	5	————		2	RxD		
RxD	6	————		3	TxD		
CTS	8	————		5	RTS		
DTR	9	————		20	DSR		

Type 3: To connect Basic/Extensive Module to a DTE with 3-pin wiring.

If [Hardware flow control] feature is set to "ON", user must loop back (or short) the RTS with CTS and DSR with DTR, DCD on MOXA site.

C320 Turbo RJ-45 Jack				DTE Device DB25 Male				C320 Turbo RJ-45 Jack				DTE Device DB9 Male			
TxD	5	———	3	RxD				TxD	5	———	2	RxD			
RxD	6	———	2	TxD				RxD	6	———	3	TxD			
GND	4/7	———	7	GND				GND	4/7	———	5	GND			
RTS	3	□						RTS	3	□					
CTS	8	□						CTS	8	□					
DTR	9	□						DTR	9	□					
DSR	2	□						DSR	2	□					
DCD	1	□						DCD	1	□					

If [Hardware flow control] feature is set to "OFF", users could just leave RTS, CTS, DSR, DTR, DCD open.

C320 Turbo RJ-45 Jack				DTE Device DB25 Male				C320 Turbo RJ-45 Jack				DTE Device DB9 Male			
TxD	5	———	3	RxD				TxD	5	———	2	RxD			
RxD	6	———	2	TxD				RxD	6	———	3	TxD			
GND	4/7	———	7	GND				GND	4/7	———	5	GND			
RTS								RTS							
CTS								CTS							
DTR								DTR							
DSR								DSR							
DCD								DCD							

B.3.3 RS-232 Test Cable Wiring for Basic/Extensive Module (Rackmount Option)

For Rackmount option, RJ45 to male DB25 RS-232 cable (CN20040) is provided for testing. The pinouts are as below:



RJ-45 Jack				DB25 Male	
DCD	1	————	8	DCD	
DSR	2	————	20	DTR	
RTS	3	————	5	CTS	
GND	4/7	————	7	GND	
TxD	5	————	3	RxD	
RxD	6	————	2	TxD	
CTS	8	————	4	RTS	
DTR	9	————	6	DSR	

B.4 Control Board to CPU/Basic Module Cable Pinouts

Type 1: The factory-supplied 2-meter cable (the one with 25 signal pins) is to used only when you are NOT using the power adapter. Pin assignments are given below for this 25-pin cable.

Warning! Do not use the factory-supplied 2-meter cable (with 25 signal pins) and power adapter at the same time, otherwise the Control Board or CPU/Basic Module may be seriously damaged.

Pin No.	Signal	Pin No.	Signal
1	GND	14	CLK-
2	DX+	15	FSX+
3	DR+	16	FSR+
4	GND	17	GND
5	GND	18	GND
6	GND	19	GND
7	-12V	20	+12V
8	+5V	21	+5V
9	+5V	22	+5V
10	+5V	23	+5V
11	CLK+	24	FSX-
12	FSR-	25	DR-
13	DX-		

Type 2: If the power is provided from the power adapter to the CPU/Basic Module, you will need to use the 10-pin cable accompanying with the power. The Control Board to CPU/Basic Module should have only 10 signal pins with the following pinouts.

Warning! Make the cable carefully and precisely according to the following pinout, otherwise you will risk damaging the Control Board and CPU/Basic Module.

Pin No.	Signal	Pin No.	Signal
2	DX+	14	CLK-
3	DR+	15	FSX+
11	CLK+	16	FSR+
12	FSR-	24	FSX-
13	DX-	25	DR-

Appendix C Extended UNIX ioctl() Commands

The following describes the syntax and usage of MOXA extended functions for both non-SVR4.x and SVR4.x UNIX. The variable `moxa_fd` is the returned file descriptor by `open()` a specific MOXA port. For example,

```
int moxa_fd;  
moxa_fd = open("/dev/ttya11", O_RDWR);
```

1. MIBUFED

This function let you know how many bytes queued in input buffer when this function is issued.

Syntax for SCO UNIX/XENIX

```
#define MIBUFED          0x401  
int                     count;    /*number of bytes queued in the buffer */  
  
ioctl(moxa_fd, MIBUFED, &count);
```

Syntax for SVR4.x UNIX/Solaris x86

```
#include                <sys/stropts.h>  
#include                <sys/sysmacros.h>  
#define MIBUFED          0x401  
struct strioc           ioc;  
int                     count;    /*number of bytes queued in the buffer */  
  
ioc.ic_cmd = MIBUFED;  
ioc.ic_timeout = 0;  
ioc.ic_len = sizeof(int);  
ioc.ic_dp = (char *)&count;  
ioctl(moxa_fd, I_STR, &ioc);
```

Note: Due to the characteristics of STREAMS driver, the returned count of bytes buffered only reflect the data buffered on MOXA board, not including the data buffered in STREAMS queue. In this case, the count is for reference only. For example, returned count may always be zero, but there still are data buffered in STREAMS queue.

2. MOBUFED

This function let you know the byte count queued in output buffer when this function is issued.

Syntax for SCO UNIX/XENIX

```
#define MOBUFED          0x402
int                      count;      /* number of bytes queued in the output buffer */

ioctl(moxa_fd, MOBUFED, &count);
```

Syntax for SVR4.x UNIX/Solaris x86

```
#include                <sys/stropts.h>
#include                <sys/sysmacros.h>
#define MOBUFED          0x402
struct striocctl        ioc;
int                      count;      /* number of bytes queued in the output buffer */

ioc.ic_cmd = MOBUFED;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&count;
ioctl(moxa_fd, I_STR, &ioc);
```

Note: See MIBUFED for influence of STREAMS driver.

3. MTCRTS

This function, only valid when hardware flow control is turned off (see MHWFLOW), is used to turn RTS on or off.

Syntax for SCO UNIX/XENIX

```
#define MTCRTS           0x403
#define TurnON           1
#define TurnOFF          0

ioctl(moxa_fd, MTCRTS, TurnON);
ioctl(moxa_fd, MTCRTS, TurnOFF);
```

Syntax for SVR4.x UNIX/Solaris x86

```

#include          <sys/stropts.h>
#include          <sys/sysmacros.h>
#define MTCRTS    0x403
#define TurnON    1
#define TurnOFF   0
struct strioctl   ioc;
int               setting;

setting = TurnON /* or TurnOFF */;
ioc.ic_cmd = MTCRTS;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);

```

4. MTCRTS

This function, only valid when hardware flow control is turned off (see MHWFLOW), is used to turn DTR on or off.

Syntax for SCO UNIX/XENIX

```

#define MTCRTS    0x404
#define TurnON    1
#define TurnOFF   0

ioctl(moxa_fd, MTCRTS, TurnON);
ioctl(moxa_fd, MTCRTS, TurnOFF);

```

Syntax for SVR4.x UNIX/Solaris x86

```

#include          <sys/stropts.h>
#include          <sys/sysmacros.h>
#define MTCRTS    0x404
#define TurnON    1
#define TurnOFF   0
struct strioctl   ioc;
int               setting;

setting = TurnON /* or TurnOFF */;
ioc.ic_cmd = MTCRTS;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);

```

5. MLOWATER

Sometimes the application software may not be able to write any further data to the output buffer because of the output buffer being full. The application has to wait until the output buffer has 'enough space' again. The criteria to tell if the output buffer has 'enough space' is that whether the output buffer reached its 'low water' level. That is the output buffer will accept further data only when the 'low water' level is reached.

If the 'low water' value is relatively small, you may find that the output buffer become empty before you write another block of data (this is quite possible because UNIX is time-sharing multitasking environment). This will result in discontinuous data transmission. In a timeout-sensitive application, e.g. facsimile (FAX) transmission, discontinuous data may alter the operation.

The default 'low water' is 512 bytes. You can enlarge it but better not exceed one half of the output buffer. Each port's output buffer depends on the number of total ports. See below.

Model	Output buffer/each port
C320 <i>Turbo</i> w/8 ports	32K bytes
C320 <i>Turbo</i> w/16ports	16K bytes
C320 <i>Turbo</i> w/24ports	8K bytes
C320 <i>Turbo</i> w/32ports	4K bytes

Syntax for SCO UNIX/XENIX

```
#define MLOWATER    0x405
int                lowater;    /* low water value of output buffer (default = 512 bytes) */

ioctl(moxa_fd, MLOWATER, lowater);
```

Syntax for SVR4.x UNIX/Solaris x86

```
#include            <sys/stropts.h>
#include            <sys/sysmacros.h>
#define MLOWATER    0x405
struct strioclt
int                lowater; /* low water value of output buffer (default = 512 bytes) */

ioc.ic_cmd = MLOWATER;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&lowater;
ioctl(moxa_fd, I_STR, &ioc);
```

6. MSTATUS

This function is used to know the RS-232 line status (CTS/DSR/DCD).

Syntax for SCO UNIX/XENIX

```
#define MSTATUS 0x407
int status; /* status = RS-232 line status */
/* bit0»CTS (1:on, 0:off) */
/* bit1»DSR (1:on, 0:off) */
/* bit2»DCD (1:on, 0:off) */

ioctl(moxa_fd, MSTATUS, &status);
```

Syntax for SVR4.x UNIX/Solaris x86

```
#define MSTATUS 0x407
#include <sys/stropts.h>
#include <sys/sysmacros.h>

struct strioc {
    int ioc;
    int status; /* status = RS-232 line status */
    /* bit0»CTS (1:on, 0:off) */
    /* bit1»DSR (1:on, 0:off) */
    /* bit2»DCD (1:on, 0:off) */

    ioc.ic_cmd = MSTATUS;
    ioc.ic_timeout = 0;
    ioc.ic_len = sizeof(int);
    ioc.ic_dp = (char *)&status;
    ioctl(moxa_fd, I_STR, &ioc);
```

7. MHWFLOW

This function is used to enable/disable hardware flow control. The first open() function of a port will set the hardware flow control bits on or off depending on the [RTS/CTS Hardware Flow Control] configuration in **moxaadm**. However, users might want to control the DTR or RTS signal on their will, thus the RTS flow control bit should be turned off (HWFlowControlOff) in order to take over the control of DTR or RTS signal via function MTCRTS or MTCDTR. MTCRTS and MTCDTR can be effective only after the RTS flow control bit is turned off.

Syntax for SCO UNIX/XENIX

```
#define MHWFLOW 0x40e
```

```

#define HWFlowControlOff      0x00
#define CTSFlowControlBitOn  0x01
#define RTSFlowControlBitOn  0x02
#define HWFlowControlOn      0x03

ioctl(moxa_fd, MHWFLOW, CTSFlowControlBitOn);
ioctl(moxa_fd, MHWFLOW, RTSFlowControlBitOn);
ioctl(moxa_fd, MHWFLOW, CTSFlowControlBitOn | RTSFlowControlBitOn);

```

Syntax for SVR4.x UNIX/Solaris x86

```

#include                <sys/stropts.h>
#include                <sys/sysmacros.h>

#define MHWFLOW         0x40e
#define HWFlowControlOff 0x00
#define CTSFlowControlBitOn 0x01
#define RTSFlowControlBitOn 0x02
#define HWFlowControlOn  0x03
struct strioctl
int                      ioc;
                        setting;

setting = CTSFlowControlbitOn;
ioc.ic_cmd = MHWFLOW;
ioc.ic_timeout = 0;
ioc.ic_len = sizeof(int);
ioc.ic_dp = (char *)&setting;
ioctl(moxa_fd, I_STR, &ioc);

```

Intellio C218/C320 Family

Customer name:	
Company:	
Tel:	Fax:
Email:	Date:

1. **Moxa Product:** ☐ C320 *Turbo* ☐ C320 ☐ C218 *Plus* ☐ C218 **Serial Number:** _____
2. **Moxa Driver Version:** _____
3. **Moxa hardware settings:** Memory Bank Address: _____ IRQ: _____
4. **Operating System:** ☐ DOS ☐ Windows 3.x ☐ Windows 95
 ☐ Windows NT 3.51 ☐ Windows NT 4.0 ☐ Windows 95 (ver. B)
 ☐ OS/2 ver ____ ☐ Linux kernel ____
 ☐ SCO UNIX 3.2.4 ☐ SCO OpenServer Release 5
 ☐ SCO Xenix ☐ (SCO) UnixWare ver ____
 ☐ Solaris x86 ver ____ ☐ SVR4.2 UNIX ☐ FreeBSD
 ☐ Others
5. **BIOS Shadow RAM enabled:** ☐ C8000 ☐ CC000 ☐ D0000
 ☐ D4000 ☐ D8000 ☐ DC000
6. **Memory management software under DOS or Win95 CONFIG.SYS:**
 ☐ EMM386 ☐ QEMM386 ☐ None
 Syntax: _____
7. **PC Host:** Make _____ Model _____
8. **CPU:** Speed _____ MHz Make _____ Model _____
9. **SCSI Card :** ☐ None ☐ Yes, memory bank address _____ IRQ _____
10. **Problem Description:** Please describe the symptom as clear as possible including the error message you see. We may have to follow your description to reproduce the symptom.
 ☐ Board not found. ☐ Board found, but can't transfer data.
 ☐ Can transfer data, but lose data. ☐ Can transfer data but with garbled data.
 ☐ Others. Detailed error message description is recommended:

RETURN PROCEDURE

For product repair, exchange or refund, the customer must:

- ✓ Provide evidence of original purchase
- ✓ Obtain a Product Return Agreement (PRA) from the sales representative or dealer
- ✓ Fill out the Problem Report Form (PRF) as detailed as possible for shorter product repair time.
- ✓ Carefully pack the product in anti-static package, and send it, pre-paid, to the dealer. The PRA should show on the outside of the package, and include a description of the problem along with the return address and telephone number of a technical contact.

